INTELLEC® MICROCOMPUTER DEVELOPMENT SYSTEM HARDWARE REFERENCE MANUAL

Manual Order No. 9800132C

intel®

INTELLEC® MICROCOMPUTER DEVELOPMENT SYSTEM HARDWARE REFERENCE MANUAL

Manual Order No. 9800132C

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

INTELLEC® MICROCOMPUTER DEVELOPMENT SYSTEM REFERENCE MANUAL

PREFACE

This reference manual is intended as the primary source of information on the hardware within the INTELLEC® MDS Microcomputer Development System. We have tried to explain, in an easy to follow format, how each of the modules within the system works, as well as provide detailed information on how to utilize each module, to its fullest extent, in the MDS System or in an independent OEM application. The reader is also referred to the "INTELLEC® MDS OPERATOR'S MANUAL" for complete instructions on how to operate the INTELLEC® MDS System.

TABLE OF CONTENTS

Chapter *Page No.*

TABLE OF CONTENTS (continued)

Chapter *Page No.*

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (continued)

LIST OF TABLES

LIST OF TABLES (continued)

Chapter 1
INTRODUCTION

The INTELLEC MDS is a complete microcomputer design center that provides total support through your entire product design cycle, from the earliest program development to the final in-circuit hardware testing and debugging of your product. Moreover, the INTELLEC MDS is a modular system, which you can custom-tailor to your own requirements. You can choose from a complete spectrum of standard modules and options.

The basic INTELLEC MDS is a complete, coordinated computer system, designed around Intel's popular 8080 Microprocessor. With the 8080, you have a 2-μs instruction cycle, a repertoire of 72 powerful instructions, unlimited subroutine nesting, and a versatile interrupt scheme. The 8080 supports up to 65,536 (64K) words of memory and up to 512 I/O devices (256 input and 256 output). But, the INTELLEC MDS System is much more than just an 8080 Microprocessor. The basic hardware configuration includes 16,384 (16K) bytes of Random-Access-Memory (RAM), and six fully-implemented I/O interfaces to:

- a Teletype (including its paper tape reader),
- a CRT terminal (or other compatible device),
- a high-speed paper tape reader,
- a high-speed paper tape punch,
- a line printer, and
- Intel's Universal PROM Programmer.

The basic system also provides an easy-to-use front panel, an 18-card chassis with etched motherboard for module interconnection, two power supplies, and a host of hardware features that includes:

- the Intel bus, which supports multi-processor configurations (8 or 16-bit), and which allows for "master-slave" relationships between modules such as those used in high-speed Direct–Memory-Access (DMA) transfers.
- an 8-level, nested interrupt priority resolution network, and
- a real-time clock with associated status bit and interrupt request line.

The basic INTELLEC MDS software package is as comprehensive as the list of hardware features. Standard INTELLEC MDS software includes a System Monitor/Debugger (firmware implemented and easily initiated by pressing a few switches on the front panel), a Macro Assembler for generating object code from symbolic macro and assembly language instructions, and a powerful Text Editor for efficient program alterations. All of these software modules execute in the INTELLEC MDS System.

The addition of INTELLEC MDS options can significantly expand the system's capabilities. You can add on additional RAM memory up to 64K words (in 16K increments). The read/write capability of RAM memory allows you to write, debug, and optimize your application routines without ever having to wait, or spend the money, for changes in metal-masked Read-Only-Memory (ROM). Or, you can add erasable, electrically Programmable Read-Only-Memory (PROM) to the system in 6K-byte increments. PROMs are ideal for storing debugged system software; less expensive than RAM but reprogrammable, unlike ROM. If you desire expanded I/O capabilities, you can acquire Intel's Input/Output Modules. Each I/O Module provides four input ports and four output ports. If your system requires high-speed direct-memory-access capabilities, Intel offers DMA Modules designed especially for use with the powerful Intel Bus (5-MHz maximum transfer rate). Each DMA Module provides five I/O ports, as well as complete bus interface logic.

You can continue to expand your system capabilities with:

- an In-Circuit Emulator (ICE), which allows you to plug the INTELLEC MDS (with all its capabilities) into your product, in place of its microprocessor, to perform final product debugging, production testing, and product verification in your system's real-time environment;
- a complete diskette system, including Intel's Diskette Operating System Software (ISIS); and
- a Universal PROM Programmer to program all of Intel's programmable-read-only-memory devices.

Chapter 2
SYSTEM OVERVIEW

The INTELLEC MDS is a complete, modular microcomputer development system. In this chapter, we identify each of the INTELLEC MDS modules and discuss, in general, how the different modules interact to provide a coordinated computer system.

The standard INTELLEC MDS System consists of four modules:

- Central Processor (CPU) Module
- Front Panel Control Module
- Monitor Module
- RAM Module (16K)

In addition, seven other modules that also plug directly into the INTELLEC MDS chassis are available as options:

- PROM Module
- Direct Memory Access (DMA) Module
- Input/Output (I/O) Module
- ICE-80 (In-Circuit Emulator for Intel 8080-based applications)
- ICE-30 (In-Circuit Emulator for Intel Series 3000-based applications)
- ROM Simulator
- Diskette Controller

Figure 2-1 illustrates the various modules within the INTELLEC MDS System. The ICE-80, ICE-30, ROM Simulator and Diskette Controller options, however, are not described in this manual. (Refer to the appropriate reference manual for each of these options.)

CENTRAL PROCESSOR (CPU) MODULE

The basic capabilities of the CPU Module are obtained through the use of Intel's 8080 Microprocessor. This processor contains an 8-bit accumulator, six 8-bit general purpose registers, and an 8-bit parallel Arithmetic and Logic Unit (ALU). Sixteen latched address lines enable the 8080 to address 65,536 bytes of external memory. As many as 256 8-bit input ports and 256 8-bit output ports may also be addressed directly. A 16-bit program counter and a 16-bit stack pointer permit flexible handling of subroutines and multi-level interrupts. The 8080's internal control logic recognizes and executes 72 different instructions. These are encoded numerically in a binary format consisting of one, two, or three 8-bit bytes. Five internal status flags enable conditional jumps, calls and returns, based on carry (overflow-underflow), sign, zero, parity, and auxiliary carry.

While the 8080 Microprocessor provides the module with an impressive set of basic processing capabilities, the module's overall performance potential is further enhanced by the remaining logic on the board. A crystal-controlled oscillator and clock generator provide a stable timing reference for all circuitry on the board. Bus control logic on the module resolves exchanges of bus control between the CPU module and other modules capable of acquiring control of the bus. The ability to resolve such exchanges makes the CPU module an ideal component in systems requiring a high-speed Direct Memory Access (DMA) capability or for systems employing a multi-processor configuration. Memory and I/O interface logic is also provided on the CPU module. The module drives a three-state, 16-line address bus, which communicates with external memory and I/O device decoding logic. A bidirectional, 8-line data bus provides the pathway for the actual data transfers. Logic on the CPU module monitors the status signals from the 8080 processor, and generates the appropriate transfer commands: MRDC (memory read), MWTC (memory write), IORC (I/O read), and IOWC (I/O write).

An 8-level, nested interrupt priority scheme rounds out the CPU module's capabilities. The interrupt logic resolves simultaneous interrupt requests on a priority basis and passes the appropriate vector to the processor, causing it to interrupt program

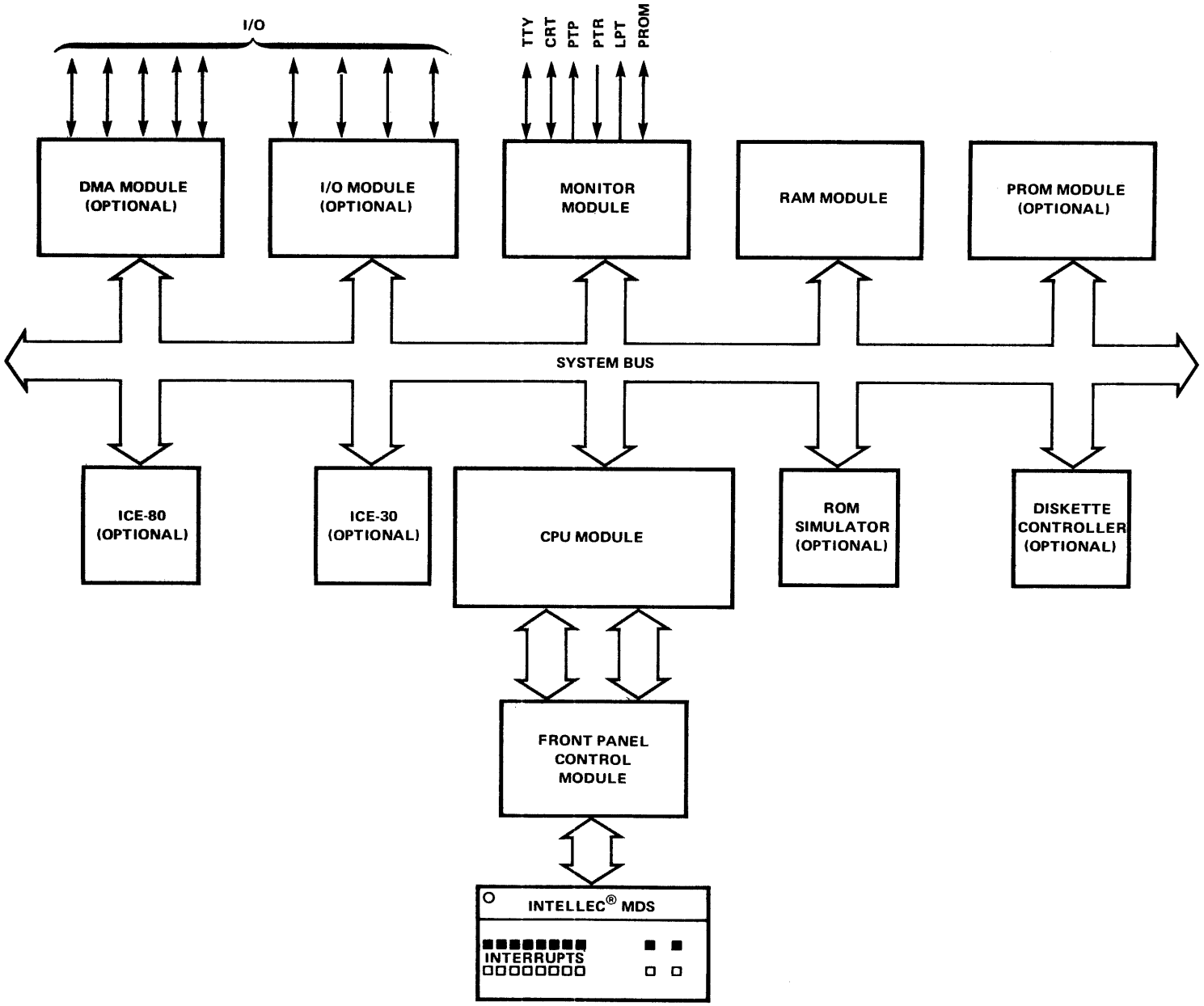


Figure 2-1. INTELLEC® MDS System Block Diagram

execution and branch to one of eight dedicated memory locations.

FRONT PANEL CONTROL MODULE

The Front Panel Control Module, as its name implies, controls the front panel in the INTELLEC MDS System. The module drives the INTERRUPT, RUN and HALT indicators, and responds to the INTERRUPT, BOOT and RESET switches. The 256-byte bootstrap program is actually stored in a PROM on the board. The module's capabilities are not, however, restricted to controlling the front panel. The module provides the system with the following additional features:

- Eight-level parallel bus priority network that resolves all requests for control of the bus, on the basis of relative priority.
- Real-time clock that sets a status bit and generates an interrupt request at 1-ms intervals; the interrupt request, however, can be disabled under program control.
- Failsafe scheme that can be used to prevent the system from stopping because a nonexistent memory location or I/O port was addressed. After waiting 10 ms, the failsafe logic generates the necessary acknowledge signal, asserts an interrupt request, and lights an indicator on the module. This feature can be very useful during program development and debugging. The acknowledge and interrupt portions can be easily disabled by disconnecting two solderless jumper pads if the features are not required.

MONITOR MODULE

The Monitor Module provides the INTELLEC MDS System with firmware storage for the Monitor program (2K words), and I/O interfaces to the following peripheral devices:

- Teletype (TTY) including paper tape reader,
- Cathode Ray Tube (CRT) terminal or other compatible device (TTL or RS232 interfaces are jumper-selectable),
- high-speed paper tape reader and punch,
- line printer, and
- PROM Programmer.

RAM MODULE

The RAM Module provides the INTELLEC MDS System with 16,384 (16K) X 8-bit words of dynamic random access memory (read/write). Up to four RAM Modules can be used in the INTELLEC MDS System, providing the system with 65,536 words of read/write memory. The RAM Module can complete a read cycle in 735 ns (worst case), and a write cycle in 1.36 μs (worst case). In addition, all of the logic required to refresh the dynamic RAM elements (at 12-μs intervals) is included on the module.

PROM MODULE

The PROM Module provides up to 6,144 (6K) X 8-bit words of PROM storage for the INTELLEC MDS System. Up to twenty-four 8702A erasable and electrically Programmable-Read-Only-Memory (PROM) devices can be included on the module. Each 8702A PROM provides 256 X 8 bits of storage. In addition, Intel's 1702 PROMs or 1302 ROMs (both are pin-compatible with the 8702A) can be used with the PROM Module. Up to 12K of PROM (or ROM) memory can be implemented in the system (i.e., two PROM Modules).

DIRECT MEMORY ACCESS (DMA) MODULE

The DMA Module provides a direct memory access capability for the high-speed transfer of data. Once a DMA operation is initiated by the Central Processor Unit (CPU), the DMA Module controls the actual transfer of up to 65,536 words of data between memory and an external device without any further intervention of the CPU required. The DMA Module can "steal" cycles by requesting control of the system bus for each word transferred. In addition, the CPU can, prior to the beginning of a transfer operation, invoke an override capability for the DMA Module. In this case, the DMA Module retains control of the bus until the entire block of data is transferred. After the entire transfer is completed, the CPU would, in response to a DMA

interrupt, reset the override capability. This mode of operation allows for "burst" mode transfers to/from very high-speed peripherals.

The DMA Module includes provisions that allow it to be interrupt-driven. In fact, the DMA interrupt request can be asserted on any one of eight interrupt priority levels. A DMA interrupt request can originate in the external device (with or without a delay), in the DMA Module itself (upon completion of a transfer operation), or can be generated by the program being executed in the CPU. The CPU program can also enable/disable interrupts or reset an existing interrupt request.

In addition to providing a high-speed data path between memory and peripheral devices, the DMA module includes five I/O ports that allow the CPU to directly address and access five devices (or groups of devices). The fifth port is associated with a 4-bit tag register. When this fifth I/O port is addressed, the contents of the tag register can be used to "steer" the input or output strobe to one of 16 additional devices, thus expanding the I/O capability of the DMA Module.

INPUT/OUTPUT (I/O) MODULE

The I/O Module includes four input and four output ports. Each output port latches 8-bit data words and issues a framed strobe pulse, of selectable duration, to the device. All outputs are driven by TTL-level buffer drivers. Each input port also supports 8 bits of data, latched or unlatched. All inputs are terminated by dual-in-line, socket-mounted resistor packs.

The I/O Module includes provisions for accepting eight external interrupt requests, buffering them and driving them on eight interrupt priority level lines. In addition, each of the eight I/O ports includes an interrupt request line that is activated by a strobe pulse from the device that is automatically cleared after the port is serviced. These port interrupt requests can be asserted on the system interrupt status register on another module.

FRONT PANEL, CABINET, MOTHERBOARD, AND POWER SUPPLIES

The INTELLEC MDS System is delivered ready-to-use, housed in a cabinet capable of holding eighteen 12-in. X 6.75-in. PCBs, with an etched motherboard that connects all of the system modules, and two power supplies that provide the necessary DC levels for system operation. The INTELLEC MDS front panel is simple but highly functional, allowing the operator to load a bootstrap program, reset the entire system, or manually initiate an interrupt request on any one of eight interrupt levels. Various indicator lights inform the operator of the current status of the system.

NOTE: All signals that appear on the INTELLEC MDS System bus are active-low. Within a module, however, both active-high and active-low signals appear. The following notation should eliminate any confusion when reading subsequent chapters: Whenever a signal is active-low, its mnemonic is followed by a slash; for example, MRDC/ means that the level on that line will be low when the memory read command is true (active). If the signal is subsequently inverted, thus making it active-high, the slash is omitted; for example, MRDC means that the level on that line will be high when the memory command is true.

Chapter 3
CENTRAL PROCESSOR MODULE

The Central Processor Module has been designed specifically to serve as the Central Processing Unit (CPU) of the INTELLEC MDS Microcomputer Development System. However, its general purpose architecture and varied capabilities permit the CPU module to serve as the primary building block for any 8-bit computer system. Thus, the CPU module, like the other INTELLEC modules, is available independently on the OEM basis. All inputs and outputs are TTL-compatible, to simplify the external interface.

The basic capabilities of the module are obtained through the use of Intel's 8080 Microprocessor. This processor contains an 8-bit accumulator, six 8-bit general purpose registers, and an 8-bit parallel Arithmetic and Logic Unit (ALU). Sixteen latched address lines enable the 8080 to address 65,536 bytes of external memory. As many as 256 8-bit input ports and 256 8-bit output ports may also be addressed directly. A 16-bit program counter and a 16-bit stack pointer permit flexible handling of subroutines and multi-level interrupts.

The 8080's internal control logic recognizes and executes 72 different instructions. These are encoded numerically in a binary format consisting of one, two, or three 8-bit bytes. Instruction categories include:

(a) register-register transfers
(b) register-memory transfers
(c) arithmetic operations, including add and subtract, with and without carry or borrow
(d) Boolean logic operations, including AND, OR, XOR
(e) decimal arithmetic
(f) input/output (I/O)
(g) stack control
(h) interrupt control
(i) register operate
(j) branch control

Five internal status flags enable conditional jumps, calls and returns, based on carry (overflow-underflow), sign, zero, parity, and auxiliary carry.

While the 8080 Microprocessor provides the module with an impressive set of basic processing capabilities, the module's overall performance potential is even further enhanced by the remaining logic on the board. A crystal-controlled oscillator and clock generator provide a stable timing reference for all circuitry on the board. The use of a 2-MHz clock permits a basic machine cycle of 2-μs for those instructions that do not reference memory during their execution.

Bus control logic on the module resolves exchanges of bus control between the CPU module and other modules capable of acquiring control of the bus. The ability to resolve such exchanges makes the CPU module an ideal component in systems requiring a high-speed Direct Memory Access (DMA) capability or for systems employing a multi-processor configuration. The resolution of bus exchanges is referred to a bus clock signal which is derived independently from the processor clock, thus allowing processors (or other "bus master" devices) of different speeds to share resources on the same bus. The bus has been designed to permit single or multiple read/write transfers at a maximum rate of 5 MHz. Such transfers, however, proceed asynchronously with respect to the bus clock; transfer speed is only dependent on the transmitting and receiving devices.

Memory and I/O interface logic is also provided on the CPU module. The module drives a three-state, 16-line address bus, which communicates with external memory and I/O device decoding logic. A bi-directional, 8-line data bus provides the pathway for the actual data transfers. Logic on the CPU module monitors the status signals from the 8080 processor, and generates the appropriate transfer commands: MRDC/ (memory read), MWTC/ (memory write), IORC/ (I/O read), and IOWC/ (I/O write). The CPU module can access up to 65,536 bytes of

memory and up to 256 input and 256 output devices (8-bit I/O addresses are duplicated on address lines 0–7 and 8–15).

An 8-level, nested interrupt priority scheme rounds out the module's capabilities. The interrupt logic resolves simultaneous interrupt requests on a priority basis and passed the appropriate vector to the processor, causing it to interrupt program execution and branch to one of eight dedicated memory locations. The interrupt vector is also saved in a nested priority table. If a request is subsequently received on a higher priority level, the vector for the new level is pushed onto the nested priority table and passed to the processor, causing it to interrupt the current service routine in order to service the higher priority request. After an interrupt service routine for a particular level is completed, the program pops the level's vector off the priority table, thus allowing the processor to resume execution of the service routine for the next lower level interrupt listed in the table. All interrupt levels can be disabled as a group, or individually, under program control.

An initialization (INIT/) input to the CPU module allows all module circuitry (except the interrupt control logic) to be reset by an external device, such as a console panel.

NOTE: Future revisions of the CPU module will utilize a programmable interrupt control device. In anticipation of this future upgrade, we recommend that you use a programmed initialization sequence to reset the interrupt logic (in addition to INIT/), as described in Section 3.4.6. This will prevent the need to modify your existing software when the future upgrade is implemented.

As a stand-alone product, the Central Processor Module is almost entirely self-contained. It requires only DC power, at levels of +5, +12 and −10 VDC.

All circuitry is mounted on a 12-in. X 6.75-in. printed circuit board. Power and most signal connections enter the module through an 86-pin, double-sided PC edge connector (0.156-in. centers). An auxiliary 60-pin, double-sided PC edge connector (0.1-in. centers) is also present for use at the designer's discretion.

In the following sections we describe the Central Processor Module in detail. The material has been organized such that with each succeeding section, the reader is exposed to information of a more detailed nature. It is hoped that this gradual approach to the material will allow the user to acquire a comprehensive understanding of the module in a single reading.

The first section introduces certain basic computer concepts which will be useful in later portions of this chapter. The next section describes the module in functional terms with the emphasis on how the various functional blocks interact to provide a flexible, but coordinated central processing unit. The third section defines the internal operation of the 8080 Microprocessor in detail. A sound understanding of the 8080 is a necessary prerequisite to examining the operations of the remaining support logic. The fourth section presents the theory of operation for all of the support logic on the board. The fifth section provides information on how to utilize the module outside of the INTELLEC MDS System. The final section lists AC and DC characteristics for signals and power inputs on the module.

3.1 THE FUNCTIONS OF A COMPUTER

This section introduces certain basic computer concepts. It provides background information and definitions which will be useful in later sections. THOSE ALREADY FAMILIAR WITH COMPUTERS MAY SKIP THIS MATERIAL, AT THEIR OPTION.

3.1.1 A TYPICAL COMPUTER SYSTEM

Though the Central Processor Module is an individual module that can perform all of the processing functions within a computer system such as the INTELLEC MDS, it cannot, by itself, produce a useful end result; the processor module must continually interact with other system components that provide such capabilities as memory storage and input/output. As a result, the discussion of any individual module must constantly refer to the activities of other modules in the same system. It is therefore very important to know something about the basic functions that must be performed in any computer system before discussing the processor module in detail.

A typical digital computer consists of:

(a) A Central Processor Unit (CPU)

(b) Memory

(c) Input/Output (I/O) ports.

The memory serves as a place to store *instructions,* the coded pieces of information that direct the activities of the CPU, and *data,* the coded pieces of information that are processed by the CPU. A group of logically related instructions stored in memory is referred to as a *program.* The CPU "reads" each instruction from memory in a logically determined sequence, and uses it to initiate processing actions. If the program sequence is coherent and logical, processing produces intelligible and useful results.

The memory is also used to store the data to be manipulated, as well as the instructions that direct that manipulation. The program must be organized such that the CPU does not read a non-instruction word when it expects to see an instruction. The CPU can rapidly access any data stored in memory, but often the memory is not large enough to store the entire data bank required for a particular application. The problem can be resolved by providing the computer with one or more *input ports.* The CPU can address these ports and input the data contained there. The addition of input ports enables the computer to receive information from external equipment (such as a paper tape reader or a floppy disk) at high rates of speed and in large volumes.

A computer also requires one or more *output ports* that permit the CPU to communicate the result of its processing to the outside world. The output may go to a display, for use by a human operator, to a peripheral device that produces "hard-copy", such as a line-printer, to a peripheral storage device, such as a floppy disk unit, or the output may constitute process control signals that direct the operations of another system, such as an automated assembly line. Like input ports, output ports are addressable. The input and output ports together permit the processor to communicate with the outside world.

The CPU unifies the system. It controls the functions performed by the other components. The CPU must be able to fetch instructions from memory, decode their binary contents and execute them. It must also be able to reference memory and I/O ports as necessary in the execution of instructions. In addition, the CPU should be able to recognize and respond to certain external control signals, such as INTERRUPT and WAIT requests. The functional units within a CPU that enable it to perform these functions are described below.

3.1.2 THE ARCHITECTURE OF A CPU

A typical Central Processor Unit (CPU) consists of the following interconnected functional units:

- Registers
- Arithmetic/Logic Unit (ALU)
- Control Circuitry

Registers are temporary storage units within the CPU. Some registers, such as the program counter and instruction register, have dedicated uses. Other registers, such as the accumulator, are for more general purpose use.

Accumulator

The accumulator usually stores one of the operands to be manipulated by the ALU. A typical instruction might direct the ALU to add the contents of some other register to the contents of the accumulator and store the result in the accumulator itself. In general, the accumulator is both a source (operand) and destination (result) register.

Often, a CPU will include a number of additional general purpose registers that can be used to store operands or intermediate data. The availability of general purpose registers eliminates the need to "shuffle" intermediate results back and forth between memory and the accumulator, thus improving processing speed and efficiency.

Program Counter (Jumps, Subroutines and the Stack)

The instructions that make up a program are stored in the system's memory. The central processor references the contents of memory, in order to determine what action is appropriate. This means that

to maintain the logical order of the program, the processor must know which location contains the next instruction.

Each of the locations in memory is numbered, to distinguish it from all other locations in memory. The number which identifies a memory location is called its *address.*

The processor maintains a counter which contains the address of the next program instruction. This register is called the *program counter.* The processor updates the program counter by adding "1" to the counter each time it fetches an instruction, so that the program counter is always current.

The programmer therefore stores his instructions in numerically adjacent addresses, so that the lower addresses contain the first instructions to be executed and the higher addresses contain later instructions. The only time the programmer may violate this sequential rule is when an instruction in one section of memory is a *jump* instruction to another section of memory.

A jump instruction contains the address of the instruction which is to follow it. The next instruction may be stored in any memory location, as long as the programmed jump specifies the correct address. During the execution of a jump instruction, the processor replaces the contents of its program counter with the address embodied in the jump. Thus, the logical continuity of the program is maintained.

A special kind of program jump occurs when the stored program *"calls"* a subroutine. In this kind of jump, the processor is required to "remember" the contents of the program counter at the time that the jump occurs. This enables the processor to resume execution of the main program when it is finished with the last instruction of the subroutine.

A *subroutine* is a program within a program. Usually it is a general-purpose set of instructions that must be executed repeatedly in the course of a main program. Routines which calculate the square, the sine, or the logarithm of a program variable are good examples of the functions often written as subroutines. Other examples might be programs designed for inputting or outputting data to a particular peripheral device.

The processor has a special way of handling subroutines, in order to insure an orderly return to the main program. When the processor receives a call instruction, it increments the program counter and stores the counter's contents in a reserved memory area known as the *stack.* The stack thus saves the address of the instruction to be executed after the subroutine is completed. Then the processor loads the address specified in the call in its program counter. The next instruction fetched will therefore be the first step of the subroutine.

The last instruction in any subroutine is a *return.* Such an instruction need specify no address. When the processor fetches a return instruction, it simply replaces the current contents of the program counter with the address on the top of the stack. This causes the processor to resume execution of the calling program at the point immediately following the original call.

Subroutines are often *nested*; that is, one subroutine will sometimes call a second subroutine. The second may call a third, and so on. This is perfectly accceptable, as long as the processor has enough capacity to store the necessary return addresses, and the logical provision for doing so. In other words, the maximum depth of nesting is determined by the depth of the stack itself. If the stack has space for storing three return addresses, then three levels of subroutines may be accommodated.

Processors have different ways of maintaining stacks. Some have facilities for the storage of return addresses built into the processor itself. Other processors use a reserved area of external memory as the stack and simply maintain a *pointer* register which contains the address of the most recent stack entry. The external stack allows virtually unlimited subroutine nesting.

In addition, if the processor provides instructions that cause the contents of the accumulator and other general purpose registers to be "pushed" onto the stack or "popped" off the stack via the address stored in the stack pointer, multi-level interrupt processing (described later in this chapter) is possible. The status of the processor (i.e., the contents of all the registers) can be saved in the stack when an interrupt is accepted and then restored after the interrupt has been serviced. This ability to save the processor's status at any given

time is possible even if an interrupt service routine, itself, is interrupted.

Instruction Register and Decoder

Every computer has a *word length* that is characteristic of that machine. A computer's word length is usually determined by the size of its internal storage elements and interconnecting paths (referred to as *busses*); for example, a computer whose registers and busses can store and transfer 8 bits of information has a characteristic word length of 8 bits and is referred to as an 8-bit parallel processor. An 8-bit parallel processor generally finds it most efficient to deal with 8-bit binary fields, and the memory associated with such a processor is therefore organized to store 8 bits in each addressable memory location. Data and instructions are stored in memory as 8-bit binary numbers, or as numbers that are integral multiples of 8 bits: 16 bits, 24 bits, and so on.

This characteristic 8-bit field is often referred to as a *byte*.

Each operation that the processor can perform is identified by a unique byte of data known as an *instruction code* or *operation code*. An 8-bit word used as an instruction code can distinguish among 256 alternative actions, more than adequate for most processors.

The processor fetches an instruction in two distinct operations. In the first, it transmits the address in its program counter to the memory. In the second, the memory returns the addressed byte to the processor. The CPU stores this instruction byte in a register known as the *instruction register*, and uses it to direct activities during the remainder of the instruction execution.

The mechanism by which the processor translates an instruction code into specific processing actions requires more elaboration than we can here afford. The concept, however, should be intuitively clear to any logic designer. The 8 bits stored in the instruction register can be decoded and used to selectively activate one of a number of output lines, in this case up to 256 lines. Each line represents a set of activities associated with execution of a particular instruction code. The enabled line can be combined coincidentally with selected timing pulses, to develop electrical signals that can then be used to initiate specific actions. This translation of code into action is performed by the *instruction decoder* and by the associated control circuitry.

An 8-bit instruction code is often sufficient to specify a particular processing action. There are times, however, when execution of the instruction requires more information than 8 bits can convey.

One example of this is when the instruction references a memory location. The basic instruction code identifies the operation to be performed, but cannot specify the object address as well. In a case like this, a 2 or 3-byte instruction must be used. Successive instruction bytes are stored in sequentially adjacent memory locations, and the processor performs two or three fetches in succession to obtain the full instruction. The first byte retrieved from memory is placed in the processor's instruction register, and subsequent bytes are placed in temporary storage, as appropriate; the processor then proceeds with the execution phase.

Address Register(s)

A CPU may use a register or register-pair to hold the address of a memory location that is to be accessed for data. If the address register is *programmable* (i.e., if there are instructions that allow the programmer to alter the contents of the register), the program can "build" an address in the address register prior to executing a *memory reference* instruction (i.e., an instruction that reads data from memory, writes data to memory or operates on data stored in memory).

Arithmetic/Logic Unit (ALU)

All processors contain an arithmetic/logic unit, which is often referred to simply as the ALU. The ALU, as its name implies, is that portion of the CPU hardware which performs the arithmetic and logical operations on the binary data.

The ALU must contain an *adder* which is capable of combining the contents of two registers in accordance with the logic of binary arithmetic. This provision permits the processor to perform arithmetic manipulations on the data it obtains from memory and from its other inputs.

Using only the basic adder, a capable programmer can write routines which will subtract, multiply, and divide, giving the machine complete arithmetic capabilities. In practice, however, most ALUs provide other built-in functions, including hardware subtraction, boolean logic operations, and shift capabilities.

The ALU contains *flag bits* which specify certain conditions that arise in the course of arithmetic and logical manipulations. Flags typically include *carry, zero, sign,* and *parity*. It is possible to program jumps which are conditionally dependent on the status of one or more flags. Thus, for example, the program may be designed to jump to a special routine, if the carry bit is set following an addition instruction.

Control Circuitry

The control circuitry is the primary functional unit within a CPU. Using clock inputs, the control circuitry maintains the proper sequence of events required for any processing task. After an instruction is fetched and decoded, the control circuitry issues the appropriate signals (to units both internal and external to the CPU) for initating the proper processing action. Often the control circuitry will be capable of responding to external signals, such as an interrupt or wait request. An *interrupt* request will cause the control circuitry to temporarily interrupt main program execution, jump to a special routine to service the interrupting device, then automatically return to the main program. A *wait* request is often issued by a memory or I/O element that operates slower than the CPU. The control circuitry will idle the CPU until the memory or I/O port is ready with the data.

3.1.3 COMPUTER OPERATIONS

There are certain operations that are basic to almost any computer. A sound understanding of these basic operations is a necessary prerequisite to examining the specific operations of a particular computer.

Timing

The activities of the central processor are cyclical. The processor fetches an instruction, performs the operations required, fetches the next instruction, and so on. An orderly sequence of events like this requires timing, and the CPU therefore requires a free-running oscillator clock which furnishes the reference for all processor actions. The combined fetch and execution of a single instruction is referred to as an *instruction cycle*. The portion of a cycle identified with a clearly defined activity is called a *state*. And the interval between pulses of the timing oscillator is referred to as a *clock period*. As a general rule, one or more clock periods are necessary to the completion of a state, and there are several states in a cycle.

Instruction Fetch

The first state(s) of any instruction cycle will be dedicated to fetching the next instruction. The CPU issues a read signal and the contents of the program counter are sent to memory, which responds by returning the next instruction word. The first byte of the instruction is placed in the instruction register. If the instruction consists of more than one byte, additional states are required to fetch one byte of the instruction. When the entire instruction is present in the CPU, the program counter is incremented (in preparation for the next instruction fetch) and the instruction is decoded. The operation specified in the instruction will be executed in the remaining states of the instruction cycle. The instruction may call for a memory read or write, an input or output and/or an internal CPU operation, such as a register-to-register transfer or an add-registers operation.

Memory Read

An instruction fetch is merely a special memory read operation that brings the instruction to the CPUs instruction register. The instruction fetched may then call for data to be read from memory into the CPU. The CPU again issues a read signal and sends the proper memory address; memory responds by returning the requested word. The data received is placed in the accumulator or one of the other general purpose registers (not the instruction register).

Memory Write

A memory write operation is similar to a read except for the direction of data flow. The CPU issues

a write signal, sends the proper memory address, then sends the data word to be written into the addressed memory location.

Wait (Memory Synchronization)

As previously stated, the activities of the processor are timed by a master clock oscillator. The clock period determines the timing of all processing activity.

The speed of the processing cycle, however, is limited by the memory's *access time*. Once the processor has sent a read address to memory, it cannot proceed until the memory has had time to respond. Many memories are capable of responding much faster than the processing cycle requires. A few, however, cannot supply the addressed byte within the minimum time established by the processor's clock.

Therefore, a processor should contain a synchronization provision, which permits the memory to request a *wait* state. When the memory receives a read or write enable signal, it places a request signal on the processor's READY line, causing the CPU to idle temporarily. After the memory has had time to respond, it frees the processor's READY line, and the instruction cycle proceeds.

Input/Output

Input and Output operations are similar to memory read and write operations with the exception that a peripheral I/O device is addressed instead of a memory location. The CPU issues the appropriate input or output control signal, sends the proper address and either receives the data being input or sends the data to be output.

Data can be input/output in either parallel or serial form. All data within a digital computer is represented in binary coded form. A binary data word consists of a group of bits; each bit is either a one or a zero. *Parallel* I/O consists of transferring all bits in the word at the same time, one bit per line. *Serial* I/O consists of transferring one bit at a time on a single line. Naturally, serial I/O is much slower but it requires considerably less hardware than does parallel I/O.

Interrupts

Interrupt provisions are included on many central processors, as a means of improving the processor's efficiency. Consider the case of a computer that is processing a large volume of data, portions of which are to be output to a printer. The CPU can output a byte of data within a single machine cycle but it may take the printer the equivalent of many machine cycles to actually print the character specified by the data byte. The CPU could then remain idle waiting until the printer can accept the next data byte. If an interrupt capability is implemented on the computer, the CPU can output a data byte then return to data processing. When the printer is ready to accept the next data byte, it can request an interrupt. When the CPU acknowledges the interrupt, it suspends main program execution and automatically branches to a routine that will output the next data byte. After the byte is output, the CPU continues with main program execution. Note that this is, in principle, quite similar to a subroutine call, except that the jump is initiated externally rather than by the program.

More complex interrupt structures are possible, in which several interrupting devices share the same processor but have different priority levels. Interruptive processing is an important feature that enables maximum utilization of a processor's capacity.

Direct Memory Access (DMA)

Another important feature that improves the throughput of a processor is the ability to perform *Direct Memory Access* (DMA) transfers.

In ordinary input and output operations, the processor itself supervises the entire data transfer. Information to be placed in memory is transferred from the input device to the processor, and then from the processor to the designated memory location. In similar fashion, information that goes from memory to output devices goes by way of the processor.

Some peripheral devices, however, are capable of transferring information to and from memory much faster than the processor itself can accomplish the transfer. If any appreciable quantity of

data must be transferred to or from such a device, then system throughput will be increased by having the device accomplish the transfer directly. The processor must temporarily suspend its operation during such a transfer, to prevent conflicts that would arise if the processor and the peripheral device attempted to use the system bus simultaneously.

3.2 FUNCTIONAL ORGANIZATION OF THE CENTRAL PROCESSOR MODULE

The Intel 8080 Central Processing Unit is the major functional element on the Central Processor Module. All of the other logic on the module supports or enhances the functions that the 8080 CPU can perform. This leads to a natural and convenient distinction between the "processor" and its "peripheral logic".

The "processor" is a complete 8-bit parallel, 8080 CPU contained in a single 40-pin dual-in-line package (see Figure 3-1). The 8080 CPU includes the following functional units:

- Arithmetic and Logic Unit (ALU)
- Register array and address logic
- Instruction register and control section
- Bidirectional, three-state data bus buffer

The 8080 CPU is fully described in Section 3.3.

The remaining logic on the Central Processor Module constitutes what we refer to as the "peripheral logic". The peripheral logic consists of the following functional blocks:

- Clock generator logic
- Bus control logic
- Data and address bus buffers and drivers
- Command generation logic (with line drivers)
- READY logic
- Interrupt logic

Figure 3-2 illustrates the interaction between the various functional blocks on the Central Processor Module.

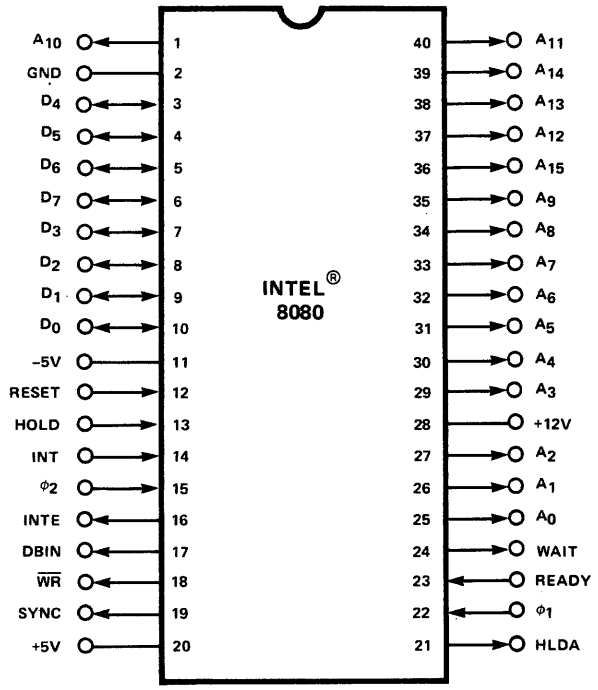


Figure 3-1. 8080 Dual-In-Line Package

The *bus control logic* allows the Central Processor Module to operate in a multi-processor configuration or, for that matter, any configuration where more than one module (e.g., processor and disk controller) can assume control of the bus. Exchanges of bus control are particularly helpful in direct memory access (DMA) transfers, where an I/O device (e.g., a disk) becomes "master" of the bus and transfers data directly to/from memory without CPU intervention.

Transitions within the bus control logic are referred to the bus clock (BCLK/). If a higher priority device is not requesting use of the bus, the Central Processor Module issues a bus request (BREQ/) signal. If no higher priority device has control, as indicated by the bus priority (BPRN/) line, the Central Processor assumes control and issues BUSY/ to inform all of the other "master" modules. In addition to BUSY/, the bus control logic generates the select (SEL) signal which

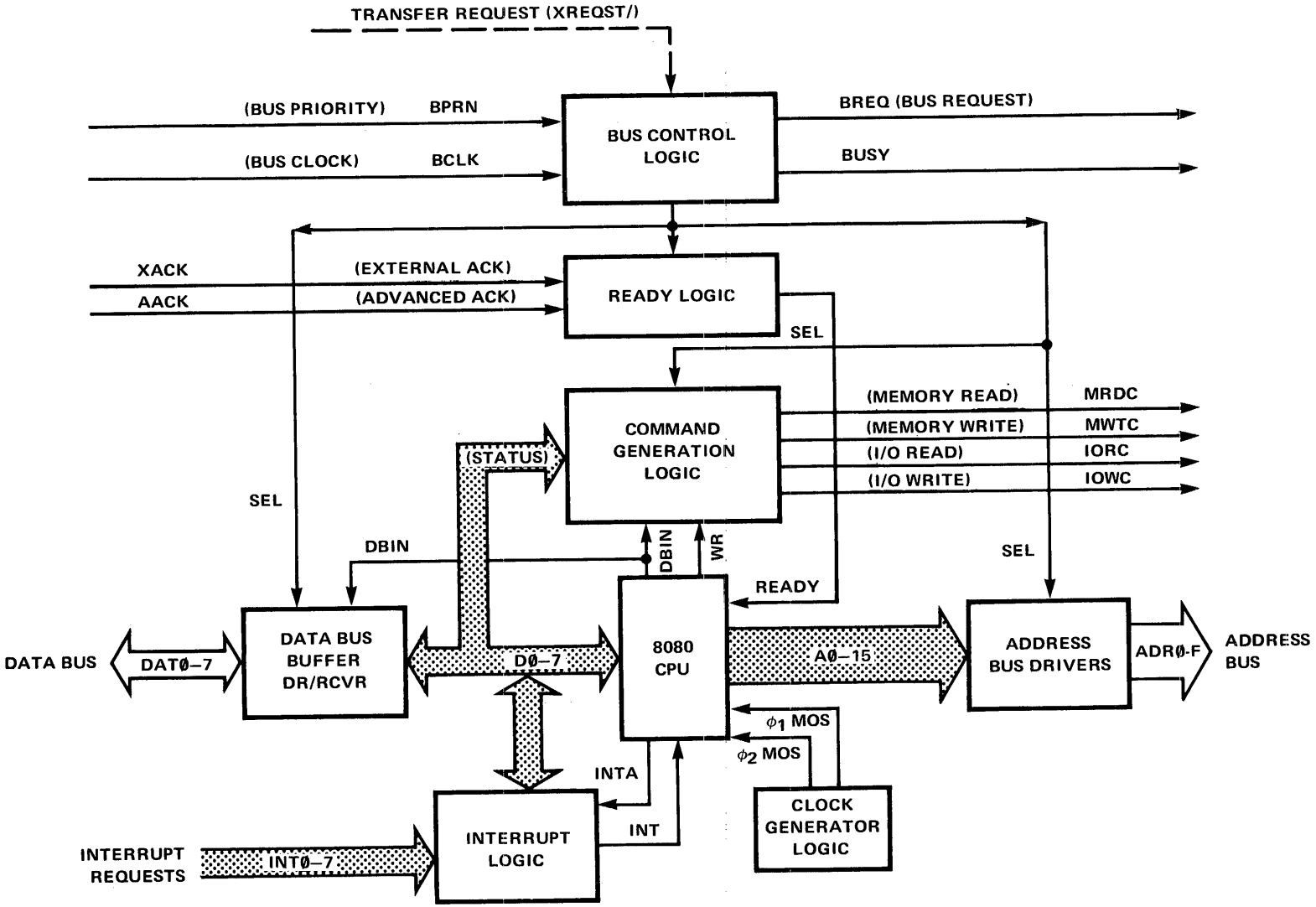


Figure 3-2. Central Processor Module Block Diagram

informs the other functional blocks on the Central Processor Module that is has control of the bus. Whenever the Central Processor Module relinquishes the bus, the absence of SEL inhibits the generation of READY. As a result, the 8080 processor idles in a wait state.

As long as the Central Processor Module has control of the bus, the processor is free to fetch and execute the program stored in external memory. All operations of the processor and the peripheral logic (except the bus control logic) are referred to two non-overlapping 2 MHz clock pulses (ϕ_1 and ϕ_2) which are produced in the *clock generation logic*. Without digressing into a detailed discussion of internal processor timing (that comes in the next section), let us for the moment merely state that the processor requires one cycle for each external access to memory or an I/O device. While the exact nature of each cycle depends on the particular operation to be performed (e.g., fetch an instruction byte or write a data byte to memory), all cycles have certain similarities. At the beginning of each cycle, the processor places an address on its address lines and places status information on its data lines.

The address uniquely identifies the "device" to be accessed, whether it be a memory location, an I/O device, or internal control logic. The address is output by the tri-state *address bus driver* circuits.

The status information on the data lines specifies the exact type of operation that is to occur during the remainder of the cycle. The *command generation logic* interprets the status bits and issues the appropriate command signal: memory read (MRDC/), memory write (MWTC/), I/O read (IORC/), interrupt (INTA/), or halt (HLTA/). The processor then removes the status bits from the data lines, thus freeing the lines for the subsequent transfer of a data byte to/from the processor.

If a particular device recognizes the address and the command, it acknowledges recognition (XACK/ is generated) and responds according to the particular command. For example, if IOWC/ is true and an output device recognizes its address on the address lines, it will generate XACK/ (which tells the processor that the device is ready to respond), and will accept the data byte that the processor has output. Memory modules also acknowledge their respective commands, MRDC/ and MWTC/, by generation of XACK/. A special acknowledge, AACK/, may be optionally used to provide an advanced indication of the module readiness to transfer data. This signal allows the 8080 CPU to avoid unnecessary wait states imposed by a conflict between the 8080 ready timing and the INTELLEC MDS Bus handshake requirements. It does, however, cause the memory timing to deviate from the INTELLEC MDS Bus specifications and must be used only in consideration of the system's absolute timing requirements (reference 3.4.4). The option is used in standard INTELLEC systems to maximize instruction processing speed but it is easily modified by use of prewired jumpers on the CPU module and memory modules that generate AACK/.

When the processor outputs data, it issues an active-low write strobe (WR/) which is used by the command generation logic. When the processor expects to input data, it issues an input strobe (DBIN).

The presence or absence of DBIN dictates direction to the bidirectional *data bus buffers*; thus enabling a data byte from the external data lines into the processor (DBIN is active) or out of the processor and onto the external data lines (DBIN is inactive).

The *interrupt logic* provides the Central Processor Module with an 8-level, nested interrupt priority capability. The logic resolves simultaneous interrupt requests on a priority basis and passes a three-bit binary encoded vector, reflecting the level currently being recognized, to the processor. The processor responds by interrupting program execution and automatically branching to one of eight dedicated memory locations. The 3-bit vector is also pushed onto a nested priority table in the interrupt logic. If an interrupt request from a higher priority level is subsequently received, the interrupt logic causes the processor to interrupt the service routine currently being executed and branch to the dedicated memory location associated with the new, higher priority level. The interrupt logic pushes the 3-bit vector for the new level onto the nested priority table. After this higher priority interrupt is serviced, the program pops its vector off the nested priority table and resumes execution of the previous, lower priority interrupted service routine.

Any of the eight interrupt levels can be individually disabled by a program-controlled interrupt

mask. In addition, all interrupts can be disabled as a group by execution of the DI instruction.

The "peripheral logic" is described in Section 3.4.

3.3 THE 8080 CENTRAL PROCESSOR UNIT

The 8080 is a complete, 8-bit parallel, Central Processor Unit (CPU) for use in general purpose digital computer systems. It is fabricated on a single LSI chip using Intel's n-channel silicon gate MOS process. The 8080 transfers data and internal state information via an 8-bit bidirectional tri-state Data Bus (D_0-D_7). Memory and peripheral device addresses are transmitted over a separate 16-bit tri-state Address Bus (A_0-A_{15}). Six timing and control outputs (SYNC, DBIN, WAIT, $\overline{WR}$, HLDA and INTE) emanate from the 8080, while four control inputs (READY, HOLD, INT and RESET), four power inputs (+12, +5, −5, and GND) and two clock inputs (ϕ_1 and ϕ_2) are accepted by the 8080.

3.3.1 ARCHITECTURE OF THE 8080 CPU

The 8080 CPU consists of the following functional units:

- Register array and address logic
- Arithmetic and Logic Unit (ALU)
- Instruction register and control section
- Bidirectional, tri-state data bus buffer

Figure 3-3 illustrates the functional blocks within the 8080 CPU.

Registers

The register section consists of a static RAM array organized into six 16-bit registers:

- Program Counter (PC)
- Stack Pointer (SP)
- Six 8-bit general purpose registers arranged in pairs, referred to as B,C; D,E; and H,L
- A temporary register pair called W,Z

The program counter maintains the memory address of the current program instruction and is incremented automatically during every instruction fetch. The stack pointer maintains the address of the next available stack location in memory. The stack pointer can be initialized to use any portion of read-write memory as a stack. The stack pointer is decremented when data is "pushed" onto the stack and incremented when data is "popped" off the stack (i.e., the stack grows "downward").

The six general purpose registers can be used either as single registers (8-bit) or as register pairs (16-bit). The temporary register pair, W,Z, are not program addressable and are only used for the internal execution of instructions.

Eight-bit data bytes can be transferred between the internal bus and the register array via the register-select multiplexer. Sixteen-bit transfers can proceed between the register array and the address latch or the incrementer/decrementer circuit. The address latch receives data from any of the three register pairs and drives the 16 address output buffers (A_0-A_{15}), as well as the incrementer/decrementer circuit. The incrementer/decrementer circuit receives data from the address latch and sends it to the register array. The 16-bit data can be incremented or decremented or simply transferred through the circuit.

Arithmetic and Logic Unit (ALU)

The ALU contains the following registers:

- An 8-bit accumulator (ACC) and a carry/link flip-flop (CY)
- An 8-bit temporary accumulator (ACT) and a temporary carry flip-flop
- A 5-bit flag register: zero, carry, sign, parity, and auxiliary carry
- An 8-bit temporary register (TEMP)

Arithmetic, logical and rotate operations are performed in the ALU. The ALU is fed by the temporary register (TMP) and the temporary accumulator (ACT) and carry flip-flop. The result of the operation can be transferred to the internal bus or to the accumulator; the ALU also feeds the flag register.

The temporary register (TMP) receives information from the internal bus and send all or portions of it to the ALU, the flag register and the internal bus.

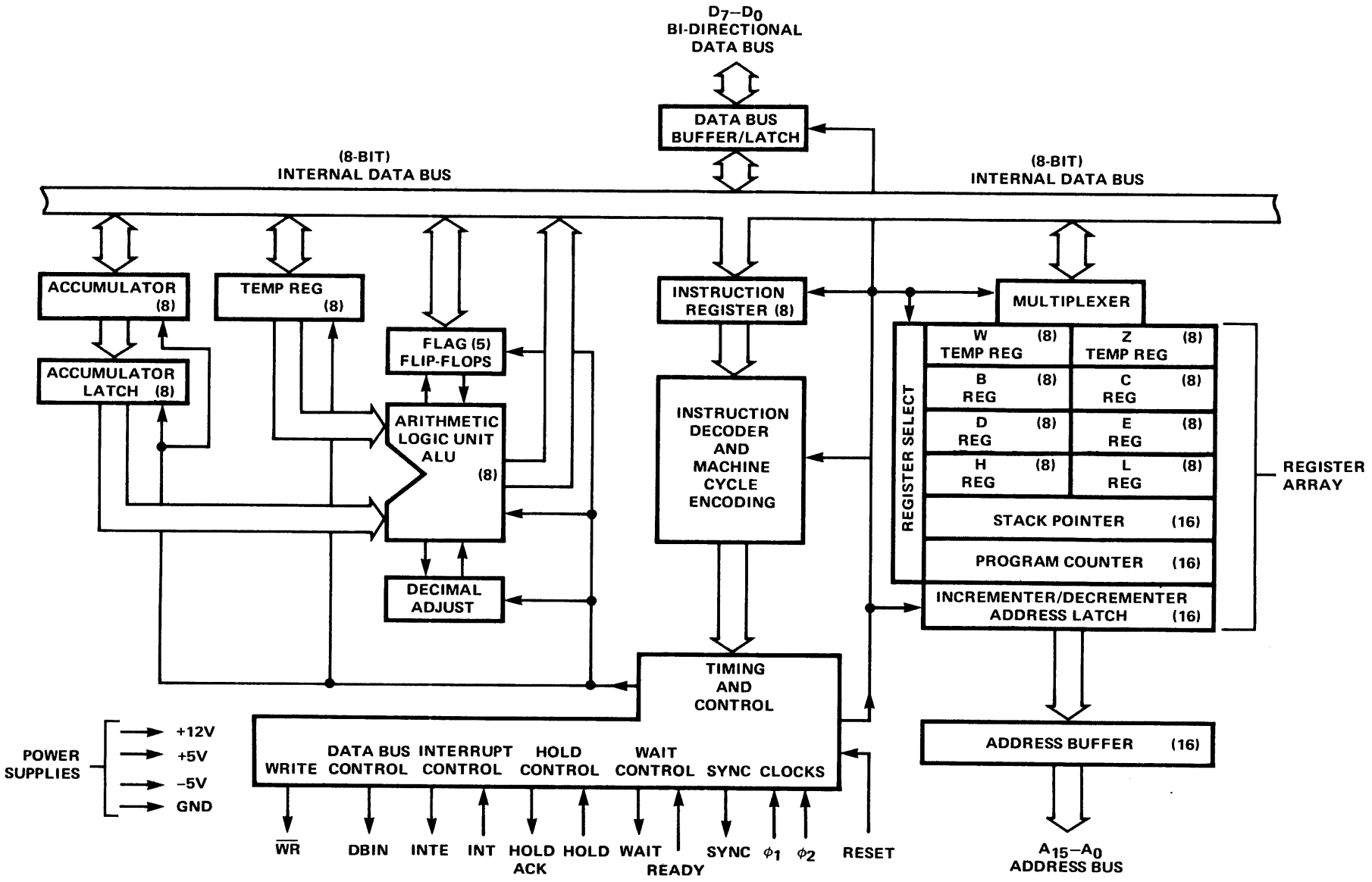


Figure 3-3. 8080 CPU Functional Block Diagram

The accumulator (ACC) can be loaded from the ALU and the internal bus, and can transfer data to the temporary accumulator (ACT) and the internal bus. The contents of the accumulator (ACC) and the auxiliary carry flip-flop can be tested for decimal correction during the execution of the Decimal Adjust Accumulator (DAA) instruction.

Instruction Register and Control

During an instruction fetch, the first byte of an instruction (containing the op code) is transferred from the internal bus to the 8-bit instruction register.

The contents of the instruction register are, in turn, available to the instruction decoder. The output of the decoder, combined with various timing signals, provides the control signals for the register array, ALU and data buffer blocks. In addition, the outputs from the instruction decoder and external control signals feed the timing and state control section which generates the state and cycle timing signals.

Data Bus Buffer

This 8-bit, bidirectional three-state buffer is used to isolate the CPUs internal bus from the external data bus (D_0 through D_7). In the output mode, the internal bus content is loaded into an 8-bit latch that, in turn, drives the data bus output buffers. The output buffers are switched off during input or non-transfer operations.

In the input mode, data from the external data bus is transferred to the internal bus. The internal bus is precharged at the beginning of each internal state, except for the transfer state (T3 – described later in this chapter).

3.3.2 THE PROCESSOR CYCLE

An *instruction cycle* is defined as the time required to fetch and execute an instruction. During the fetch, a selected instruction (one, two, or three bytes) is extracted from memory and deposited in the CPU's instruction register. During the execution part, the instruction is decoded and translated into specific processing activities.

Every instruction cycle consists of one, two, three, four, or five machine cycles. A *machine cycle* is required each time the CPU accesses memory or an I/O port. The fetch portion of an instruction cycle requires one machine cycle for each byte to be fetched. The duration of the execution portion of the instruction cycle depends on the kind of instruction that has been fetched. Some instructions do not require any machine cycles other than those necessary to fetch the instruction; other instructions, however, require additional machine cycles to write or read data to/from memory or I/O devices. The DAD instruction is an exception in that it requires two additional machine cycles to complete an internal register-pair add.

Each machine cycle consists of three, four, or five states. A *state* is the smallest unit or processing activity and is defined as the interval between two successive positive-going transitions of the ϕ_1 clock pulse. The 8080 is driven by a 2-phase clock oscillator. All processing activities are referred to the period of this clock. The two non-overlapping clock pulses, labeled ϕ_1 and ϕ_2, are furnished by external circuitry. It is the ϕ_1 clock pulse which divides each machine cycle into states. Timing logic within the 8080 uses the clock inputs to produce a SYNC pulse, which identifies the beginning of every machine cycle. The SYNC pulse is triggered by the low-to-high transition of ϕ_2, as shown in Figure 3-4.

There are three exceptions to the defined duration of a state. They are the WAIT state, the hold (HLDA) state, and the halt (HALTA) state, described later in this chapter. Because the WAIT, the HLDA, and the HALTA states depend upon external events, they are by their nature of indeterminate length. Even these exceptional states, however, must be synchronized with the pulses of the driving clock. Thus the duration of all states, including these, are integral multiples of the clock pulse.

To summarize then, each *clock period*, marks a *state*; three to five *states* summarize a *machine cycle*; and one to five *machine cycles* comprise an *instruction cycle*. A full instruction cycle requires anywhere from four to 17 states for its completion, depending on the kind of instruction involved.

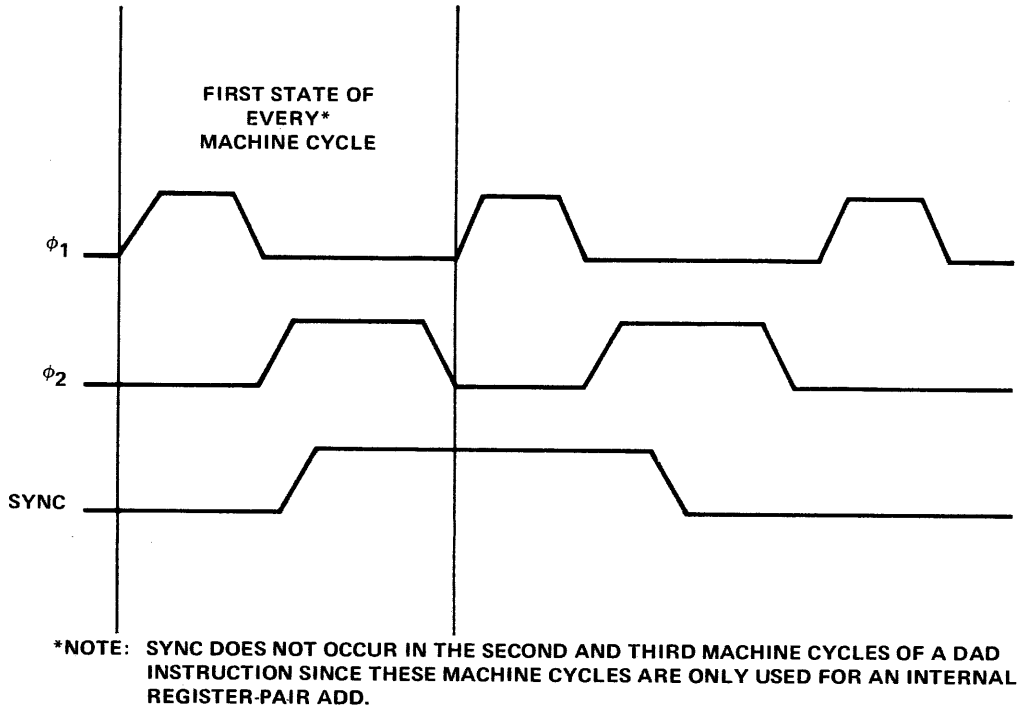


Figure 3-4. ϕ_1, ϕ_2 and Sync Timing

Machine Cycle Identification

With the exception of the DAD instruction, there is just one consideration that determines how many machine cycles are required in any given instruction cycle: the number of times that the processor must reference a memory address, or an addressable peripheral device, in order to fetch and execute the instruction. Like many processors, the 8080 is so constructed that it can transmit only one address per machine cycle. Thus, if the fetching and execution of an instruction requires two memory references, then the instruction cycle associated with that instruction consists of two machine cycles. If five such references are called for, then the instruction cycle contains five machine cycles.

Every instruction cycle has at least one reference to memory, during which the instruction is fetched. An instruction cycle must always have a fetch, even if the execution of instruction requires no further references to memory. The first machine cycle in every instruction cycle is therefore a FETCH. Beyond that, there are no fast rules. It depends on the kind of instruction.

Consider some examples. The add-register (ADD r) instruction is an instruction that requires only a single machine cycle (FETCH) for its completion. In this 1-byte instruction, the contents of one of the CPUs six general purpose registers is added to the pre-existing contents of the accumulator. Since all the information necessary to execute the command is contained in the 8 bits of the instruction code, only one memory reference is necessary: that actually used to fetch the instruction. Three states are used to extract the instruction from memory, and one additional state is used to accomplish the desired addition. The entire instruction cycle thus requires only one machine cycle that consists of four states, or four periods of the external clock.

Support now, however, that we wish to add the contents of a specific memory location to the pre-existing contents of the accumulator (ADD M). Although this is quite similar in principle to the example just cited, several additional steps will be necessary. An extra machine cycle will be needed, in order to address the desired memory location.

The actual sequence is as follows: First the processor extracts from memory the 1-byte instruction

word addressed by its program counter. This takes three states. The 8-bit instruction word obtained during the FETCH machine cycle is deposited in the CPU's instruction register and used to direct activities during the remainder of the instruction cycle. Next, the processor sends out as an address the contents of its H and L registers. The 8-bit data word returned during this MEMORY READ machine cycle is placed in a temporary register inside the 8080 CPU. By now, three more clock periods (states) have elapsed. In the seventh and final state, the contents of the temporary register are added to those of the accumulator. Two machine cycles, consisting of seven states in all, complete the "ADD M" instruction cycle.

At the opposite extreme is the save H and L registers (SHLD) instruction, which requires five machine cycles. During an "SHLD" instruction cycle, the contents of the processor's H and L registers are deposited in two sequentially adjacent memory locations; the destination is indicated by two address bytes which are stored in the two memory locations immediately following the operation code byte. The following events occur:

(1) A FETCH machine cycle, consisting of four states. During the first three states of this machine cycle, the processor fetches the instruction indicated by its program counter. The program counter is then incremented. In the fourth state, the contents of the H and L registers are transferred to temporary registers within the chip, W and Z, respectively. Data previously held in the H and L registers is thus saved, thereby clearing H and L to receive incoming data.

(2) A MEMORY READ machine cycle, consisting of three states. During this machine cycle, the byte indicated by the program counter is extracted from memory and placed in the processor's L register. The program counter is incremented again.

(3) Another MEMORY READ machine cycle, consisting of three states, in which the byte indicated by the processor's program counter is deposited in the H register. The program counter is incremented, in anticipation of the next instruction fetch.

(4) A MEMORY WRITE machine cycle, of three states, in which the contents of the Z register are transferred to the memory location pointed to by the present contents of the H and L registers. The state following the transfer is used to increment the H and L pointers, so that they indicate the next memory location to receive data.

(5) A MEMORY WRITE machine cycle, of three states, in which the contents of the W register are transferred to the new memory location pointed to by the H and L registers.

The "SHLD" instruction cycle contains five machine cycles and takes 16 states to execute.

Most instructions fall somewhere between the extremes typified by the "ADD r" and the "SHLD" instructions. The input (INP) and the output (OUT), for example, require three machine cycles: a FETCH, to obtain the instruction; a MEMORY READ, to obtain the address of the object peripheral; and an INPUT or an OUTPUT machine cycle, to complete the transfer.

There are nine types of machine cycles that may occur within an instruction cycle; though no one instruction cycle will consist of more than five machine cycles:

(a) FETCH
(b) MEMORY READ
(c) MEMORY WRITE
(d) STACK READ
(e) STACK WRITE
(f) INPUT
(g) OUTPUT
(h) INTERRUPT
(i) HALT

The machine cycles that actually do occur in a particular instruction cycle depend upon the kind of instruction, with the overriding stipulation that the the first machine cycle in any instruction cycle is always a FETCH.

The processor identifies the machine cycle in progress, by transmitting an 8-bit status signal during the first state of every machine cycle. Updated

status information is published on the 8080's data lines (D_0-D_7), during the SYNC interval. This data should be saved in latches, decoded, and used to develop control signals for external circuitry. Table 3-1 shows how the positive-true status information is distributed on the processor's data bus.

Status signals are provided principally for the control of external circuitry. Simplicity of interface, rather than machine cycle identification, dictates the logical definition of individual status bits. You will, therefore, observe that certain processor machine cycles are uniquely identified by a single status bit, but that others are not. The M_1 status bit (D_5), for example, unambiguously identifies a FETCH machine cycle. A STACK READ, on the other hand, is indicated by the coincidence of STACK and MEMR signals. Machine cycle identification data can also be valuable in the test and debugging phases of system development. Table 3-2 lists the status bit outputs for each type of machine cycle.

State Transition Sequence

Every machine cycle within an instruction cycle consists of three to five active states (referred to as T1, T2, T3, T4, T5, or TW). The actual number of states depends upon the instruction being executed, and on the particular machine cycle within the greater instruction cycle. The state transition diagram in Figure 3-5 shows how the 8080 proceeds from state to state in the course of a machine cycle. The diagram also shows how the READY, HOLD, and INTERRUPT lines are sampled during the machine cycle, and how the conditions on these lines may modify the basic transition sequence. In the present discussion, we are concerned only with the basic sequence and with the READY function. HOLD and INTERRUPT functions will be discussed later.

The 8080 CPU does not indicate its internal state directly, by transmitting a "state control" output during each state; instead, the 8080 supplies direct control output (INTE, HLDA, DBIN, $\overline{WR}$, and WAIT) for use by external circuitry.

Recall that the 8080 passes through at least three states in every machine cycle, with each state defined by successive low-to-high transitions of the ϕ_1 clock. Figure 3-6 shows the timing relationships in a typical FETCH machine cycle. Events that occur in each state are referred to transitions of the ϕ_1 and ϕ_2 clock pulses.

The SYNC signal identifies the first state (T1) in every machine cycle. As shown in Figure 3-6, the SYNC signal is related to the leading edge of the ϕ_2 clock. Status information is displayed on D_0-D_7 during this same interval. Switching of the status signals is likewise controlled by ϕ_2.

The rising edge of ϕ_2 during T1 also loads the processor's address lines (A_0-A_{15}). These lines become stable within a brief delay (t_{DA}) of the ϕ_2 clocking pulse, and they remain stable until the first ϕ_2 pulse after state T3. This gives the processor ample time to read the data returned from memory.

Once the processor has sent an address to memory, there is an opportunity for the memory to request a WAIT. This it does by pulling the processor's READY line low, prior to the "Ready set-up" interval (t_{RS}) which occurs during the ϕ_2 pulse within state T2 or TW. As long as the READY line remains low, the processor will idle, giving the memory time to respond to the addressed data request. (Refer to Figure 3-6.)

The processor responds to a wait request by entering an alternative state (TW) at the end of T2, rather than proceeding directly to the T3 state. Entry into the T_W state is indicated by a WAIT signal from the processor, acknowledging the memory's request. A low-to-high transition on the WAIT line is triggered by the rising edge of the ϕ_1 clock and occurs within a brief delay (t_{DC}) of the actual entry into the T_W state.

A wait period may be of indefinite duration. The processor remains in the waiting condition until its READY line again goes high. A READY indication must precede the falling edge of the ϕ_2 clock by a specified interval (t_{RS}), in order to guarantee an exit from the T_W state. The cycle may then proceed, beginning with the rising edge of the next ϕ_1 clock. A WAIT interval will therefore consist of an integral number of T_W states and will always be a multiple of the clock period.

The events that take place during the T3 state are determined by the kind of machine cycle in progress. In a FETCH machine cycle, the processor

Table 3-1

8080 STATUS BIT DEFINITIONS

SYMBOLS	DATA BIT BUS	DEFINITION
HLTA	D_3	Acknowledge signal for HALT instruction.
INTA	$D_\emptyset$	Acknowledge signal for INTERRUPT request. Signal should be used to gate a re-start instruction onto the data bus when DBIN is active.
INP	D_6	Indicates that the address bus contains the address of an input device and the input data should be placed on the data bus when DBIN is active.
OUT	D_4	Indicates that the address bus contains the address of an output device and the data bus will contain the output data when $\overline{WR}$ is active.
MEMR	D_7	Designates that the data bus will be used for memory read data.
M_1	D_5	Provides a signal to indicate that the CPU is in the fetch cycle for the first byte of an instruction.
STACK	D_2	Indicates that the address bus holds the pushdown stack address from the Stack Pointer.
$\overline{WO}$	D_1	Indicates that the operation in the current machine cycle will be a WRITE memory or OUTPUT function ($\overline{WO} = \emptyset$). Otherwise, a READ memory or INPUT operation will be executed.

Table 3-2

STATUS BIT DECODING

TYPE OF MACHINE CYCLE	STATUS BITS							
	$D_\emptyset$ INTA	D_1 $\overline{WO}$	D_2 STACK	D_3 HLTA	D_4 OUT	D_5 M_1	D_6 INP	D_7 MEMR
FETCH	0	1	0	0	0	1	0	1
MEMORY READ	0	1	0	0	0	0	0	1
MEMORY WRITE	0	0	0	0	0	0	0	0
STACK READ	0	1	1	0	0	0	0	1
STACK WRITE	0	0	1	0	0	0	0	0
INPUT	0	1	0	0	0	0	1	0
OUTPUT	0	0	0	0	1	0	0	0
INTERRUPT	1	1	0	0	0	1	0	0
HALT	0	1	0	1	0	0	0	0

*NOTE: 1 = high level; $\emptyset$ = low level. Notice that the write/output bit ($\overline{WO}$) is negative-true.

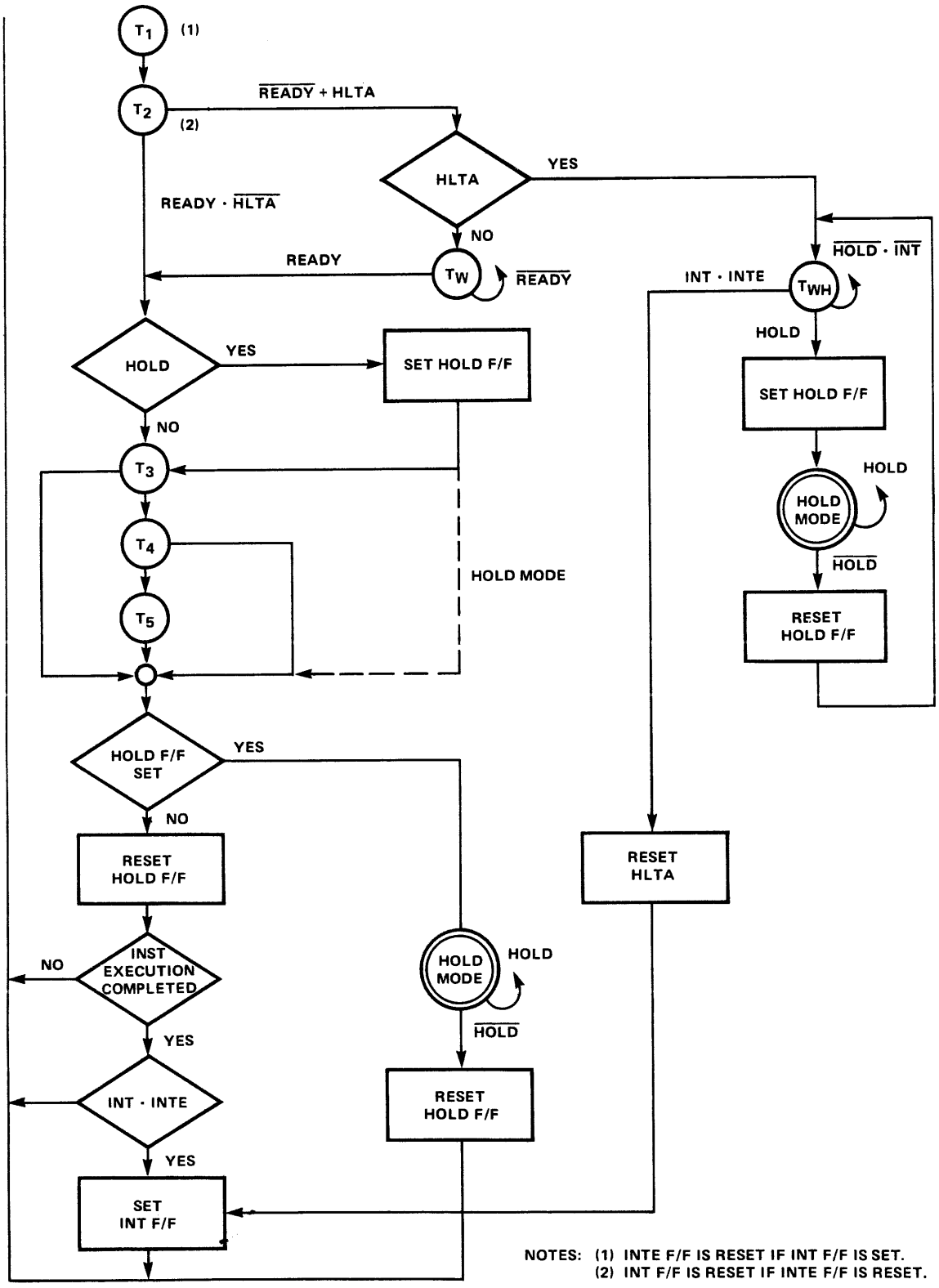


Figure 3-5. CPU State Transition Diagram

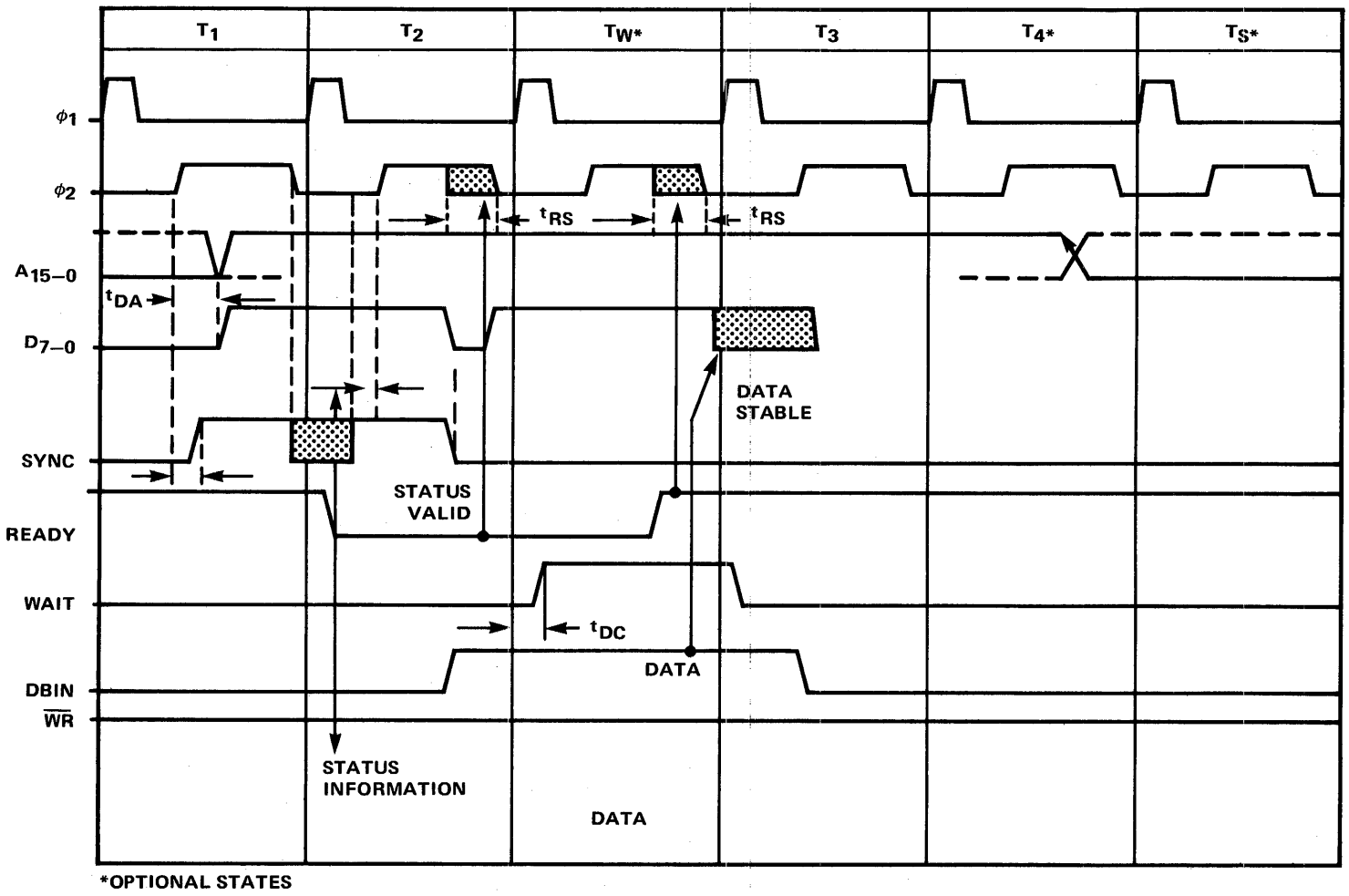


Figure 3-6. Typical FETCH Machine Cycle

interprets the data on its data bus as an instruction. During a MEMORY READ or a STACK READ, signals on the same bus are intrepreted as a data word. The processor itself outputs data on this bus during a MEMORY WRITE machine cycle. And, during I/O operations, the processor may either transmit or receive data, depending on whether an INPUT or an OUTPUT operation is involved.

Figure 3-7 illustrates the timing that is characteristic of a data input operation. As shown, the low-to-high transition of ϕ_2 during T2 clears status information from the processor's data lines, preparing these lines for the receipt of incoming data. The data presented to the processor must have stabilized, prior to both the "ϕ_1-data set-up" interval (t_{DS1}), that precedes the falling edge of the ϕ_1 pluse defining state T3, and the "ϕ_2-data set-up" interval (t_{DS2}), that precedes the rising edge of ϕ_2 in state T3. And, this same data must remain stable during the "data hold" interval (t_{DH}) that occurs following the rising edge of the next ϕ_2 pulse. Data placed on these lines by memory or by other external devices will be sampled during T3.

During the input of data to the processor, the 8080 generates a DBIN signal which should be used externally to enable the transfer. Machine cycles in which DBIN is available include: FETCH, MEMORY READ, STACK READ, and INTERRUPT. DBIN is initiated by the rising edge of ϕ_2 during state T2 and terminated by the corresponding edge of ϕ_2 during T3. Any T_W phases intervening between T2 and T3 will therefore extend DBIN by one or more clock periods.

Figure 3-8 shows the timing of machine cycles in which the processor outputs data. Output data may be destined either for memory or for peripherals. The rising edge of ϕ_2 within state T2 clears status information from the CPU's data lines, and loads in the data which is to be output to external devices. This substitution takes place within the "data output delay" interval (t_{DD}) following the ϕ_2 clock's leading edge. Data on the bus remains stable throughout the remainder of the machine cycle, until replaced by updated status information in the subsequent T1 state. Observe that a READY signal is necessary for completion of an OUTPUT machine cycle. Unless such an indication is present, the processor enters the T_W state, following the T2 state. Data on the output lines remains stable in the interim, and the processing cycle will not proceed until the READY line again goes high.

The 8080 CPU generates a WR/output for the synchronization of external transfers, during those machine cycles in which the processor outputs data. These include MEMORY WRITE, STACK WRITE, and OUTPUT. The negative-going leading edge of WR/ is referred to the rising edge of the first ϕ_1 clock pulse following T2, and occurs within a brief delay (t_{DC}) of that event. WR/ remains low until re-triggered by the leading edge of ϕ_2, during the state following T3. Note that any T_W states inserted during WR/, affect WR/ in much the same way that DBIN is affected during data input operations.

All processor machine cycles consist of at least three states: T1, T2, and T3, as just described. If the processor has to wait for a READY response, then the machine cycle may also contain one or more T_W states. During the three basic states, data is transferred to or from the processor.

After the T3 state, however, it becomes difficult to generalize. T4 and T5 states are available, if the execution of a particular instruction requires them. But not all machine cycles make use of these states. It depends upon the kind of instruction being executed, and on the particular machine cycle within the instruction cycle. The processor will terminate any machine cycle as soon as its processing activities are completed, rather than proceeding through the T4 and T5 states every time. Thus, the 8080 may exit a machine cycle following the T3, the T4, or the T5 state, and proceed directly to the T1 state of the next machine cycle.

Table 3-3 lists the general activities associated with each state. Table 3-4 summarizes the state sequencing involved in the execution of each particular type of 8080 instruction; you should refer to Table 3-4 if you have any questions on how a specific instruction is executed.

3.3.3 INTERRUPT SEQUENCES

The 8080 has the built-in capacity to handle external interrupt requests. A peripheral device can initiate an interrupt simply by driving the processor's interrupt (INT) line high.

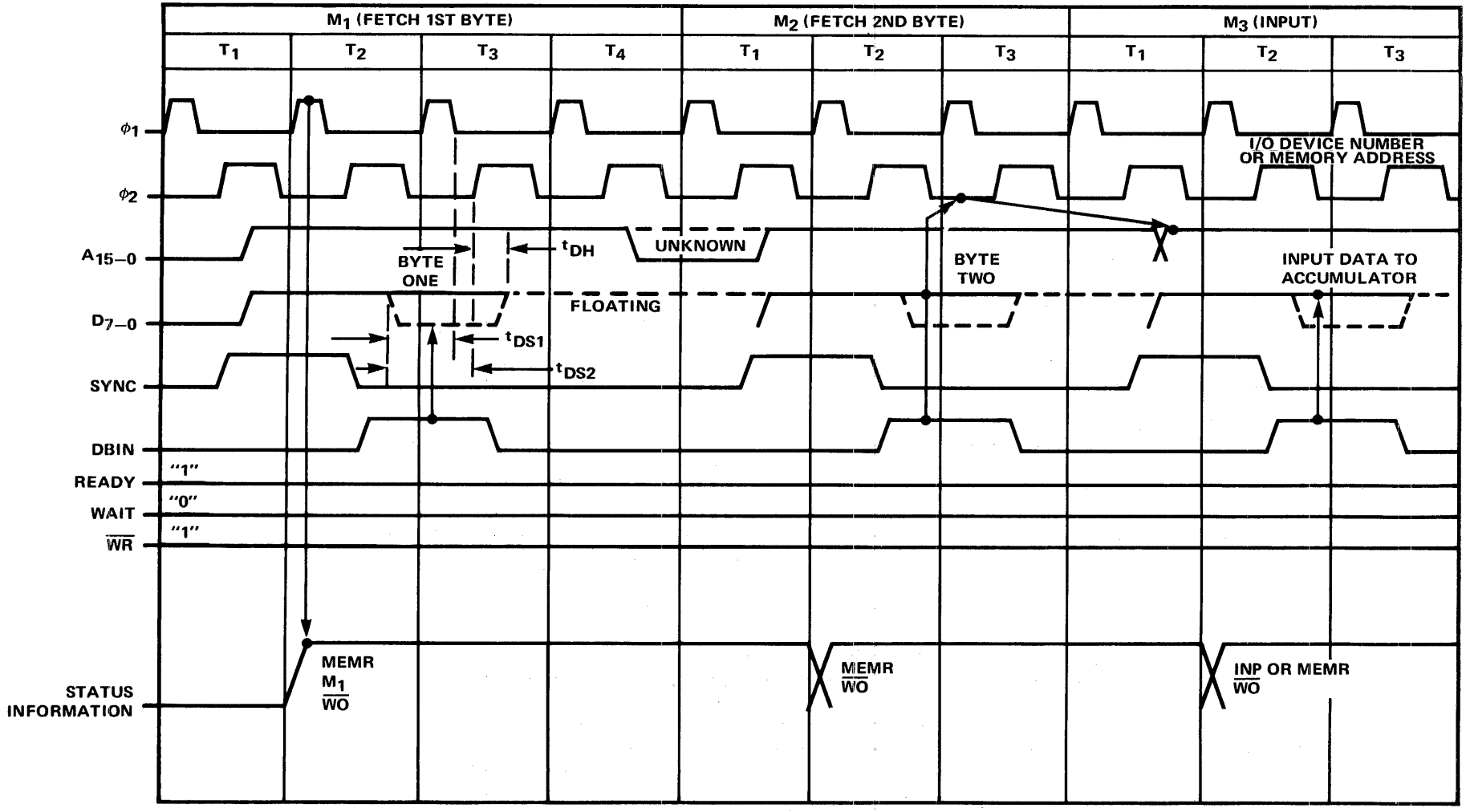


Figure 3-7. Input Instruction Cycle

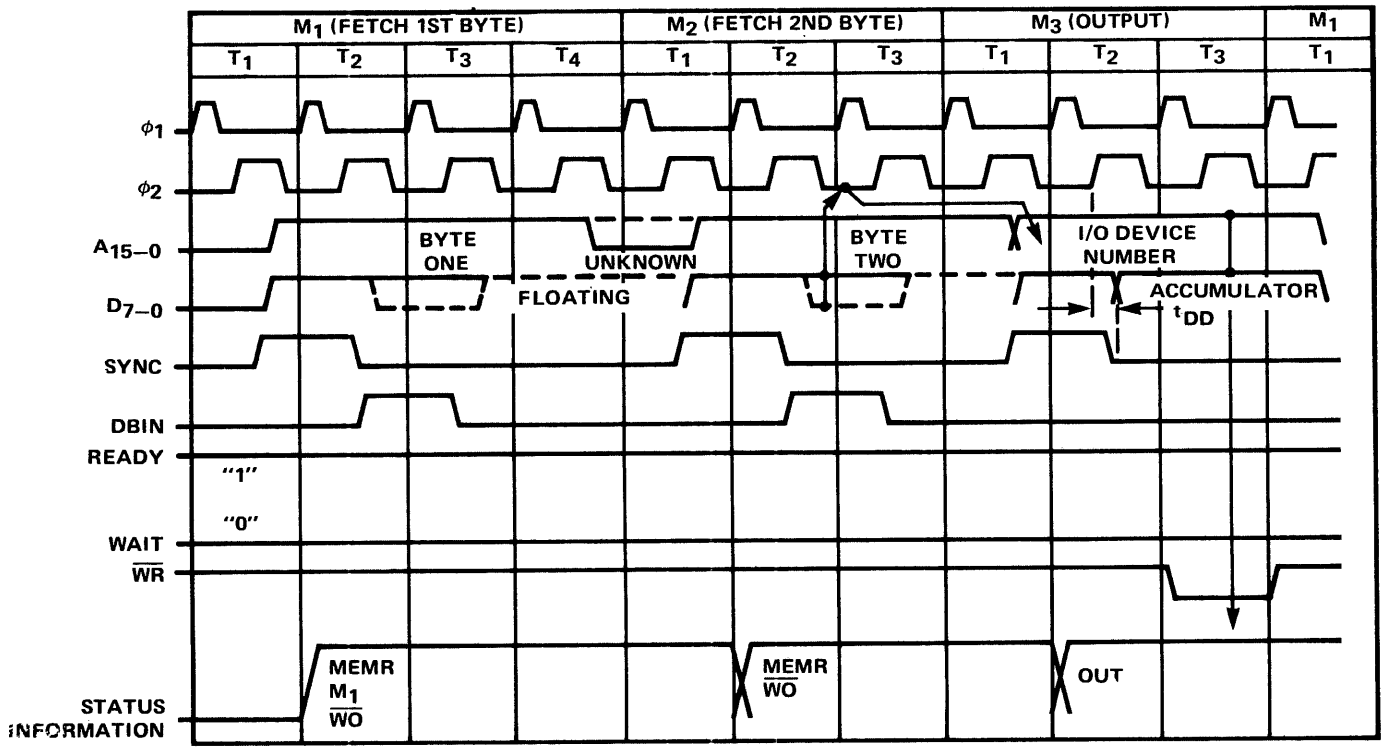


Figure 3-8. Output Instruction Cycle

Table 3-3

STATE DEFINITIONS

STATE	ASSOCIATED ACTIVITIES
T_1	A memory address or I/O device number is placed on the Address Bus ($A_{15-\emptyset}$); status information is placed on Data Bus ($D_{7-\emptyset}$).
T_2	The CPU samples the READY and HOLD inputs and checks for halt instruction.
T_W (optional)	Processor enters wait state if READY is low or if HALT instruction has been executed.
T_3	An instruction byte (FETCH machine cycle), data byte (MEMORY READ, STACK READ or INPUT machine cycle), or interrupt instruction (INTERRUPT machine cycle) is input to the CPU from the Data Bus; or a data byte (MEMORY WRITE, STACK WRITE or OUTPUT machine cycle) is output onto the data bus.
T_4 T_5 (optional)	States T_4 and T_5 are available if the execution of a particular instruction requires them; if not, the CPU may skip one or both of them. T_4 and T_5 are only used for internal processor operations.

The interrupt (INT) input is asynchronous, and a request may therefore originate at any time during any instruction cycle. Internal logic re-clocks the external request, so that a proper correspondence with the driving clock is established. As Figure 3-9 shows, an interrupt request (INT) arriving during the time that the interrupt enable line (INTE) is high, acts in coincidence with the ϕ_2 clock to set the internal interrupt latch. This event takes place during the last state of the instruction cycle in which the request occurs, thus ensuring that any instruction in progress is completed before the interrupt can be processed.

The INTERRUPT machine cycle which follows the arrival of an enabled interrupt request resembles an ordinary FETCH machine cycle in most respects. The M_1 status bit is transmitted as usual during the SYNC interval. It is accompanied, however, by an INTA status bit (D_0) which acknowledges the external request. The contents of the program counter are latched onto the CPU's address lines during T1, but the counter itself is not incremented during the INTERRUPT machine cycle, as it otherwise would be. In this way, the pre-interrupt status of the program counter is preserved, so that data in the counter may be saved in the stack. This in turn permits an orderly return to the interrupted program after the interrupt request has been processed.

The interrupt cycle is otherwise indistinguishable from an ordinary FETCH machine cycle. The processor itself takes no further special action. It is the responsibility of the peripheral logic to see that an 8-bit interrupt instruction is "jammed" onto the processor's data bus during state T3. In a typical system, this means that the data-in bus from memory must be temporarily disconnected from the processor's main data bus, so that the interrupting device can command the main bus without interference.

The 8080's instruction set provides a special 1-byte call which facilitates the processing of interrupts (the ordinary program call takes 3 bytes). This is the restart instruction (RST). A variable 3-bit field embedded in the 8-bit field of the RST enables the interrupting device to direct a jump to one of eight fixed memory locations. The decimal addresses of these dedicated locations are: 0, 8, 16, 24, 32, 40, 48, and 56. Any of these addresses may be used to store the first instruction(s) of a routine designed to service the requirements of an interrupting device.

Table 3-4

CYCLE AND STATE TRANSITION SEQUENCES FOR EACH 8080 INSTRUCTION

MNEMONIC	OP CODE		M1[1]					M2		
	$D_7 D_6 D_5 D_4$	$D_3 D_2 D_1 D_0$	T1	T2[2]	T3	T4	T5	T1	T2[2]	T3
MOV r1, r2	0 1 D D	D S S S	PC OUT STATUS	PC = PC +1	INST→TMP/IR	(SSS)→TMP	(TMP)→DDD			
MOV r, M	0 1 D D	D 1 1 0	↑	↑	↑	X[3]		HL OUT STATUS[6]	DATA→	→DDD
MOV M, r	0 1 1 1	0 S S S				(SSS)→TMP		HL OUT STATUS[7]	(TMP)→	→DATA BUS
SPHL	1 1 1 1	1 0 0 1				(HL) →	SP			
MVI r, data	0 0 D D	D 1 1 0				X		PC OUT STATUS[6]	B2→	→DDDD
MVI M, data	0 0 1 1	0 1 1 0				X		↑	B2→	→TMP
LXI rp, data	0 0 R P	0 0 0 1				X			PC = PC + 1 B2→	→r1
LDA addr	0 0 1 1	1 0 1 0				X			PC = PC + 1 B2→	→Z
STA addr	0 0 1 1	0 0 1 0				X			PC = PC + 1 B2→	→Z
LHLD addr	0 0 1 0	1 0 1 0				X			PC = PC + 1 B2→	→Z
SHLD addr	0 0 1 0	0 0 1 0				X		PC OUT STATUS[6]	PC = PC + 1 B2→	→Z
LDAX rp[4]	0 0 R P	1 0 1 0				X		rp OUT STATUS[6]	DATA→	→A
STAX rp[4]	0 0 R P	0 0 1 0				X		rp OUT STATUS[7]	(A)→	→DATA BUS
XCHG	1 1 1 0	1 0 1 1				(HL)↔(DE)				
ADD r	1 0 0 0	0 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)+(TMP)→A	
ADD M	1 0 0 0	0 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA→	→TMP
ADI data	1 1 0 0	0 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2→	→TMP
ADC r	1 0 0 0	1 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)+(TMP)+CY→A	
ADC M	1 0 0 0	1 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA→	→TMP
ACI data	1 1 0 0	1 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2→	→TMP
SUB r	1 0 0 1	0 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)-(TMP)→A	
SUB M	1 0 0 1	0 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA→	→TMP
SUI data	1 1 0 1	0 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2→	→TMP
SBB r	1 0 0 1	1 S S S				(SSS)→TMP (A)→ACT		[9]	(ACT)-(TMP)-CY→A	
SBB M	1 0 0 1	1 1 1 0				(A)→ACT		HL OUT STATUS[6]	DATA→	→TMP
SBI data	1 1 0 1	1 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1 B2→	→TMP
INR r	0 0 D D	D 1 0 0				(DDD)→TMP (TMP) + 1→ALU	ALU→DDD			
INR M	0 0 1 1	0 1 0 0				X		HL OUT STATUS[6]	DATA→ (TMP)+1→	→TMP →ALU
DCR r	0 0 D D	D 1 0 1				(DDD)→TMP. (TMP)+1→ALU	ALU→DDD			
DCR M	0 0 1 1	0 1 0 1				X		HL OUT STATUS[6]	DATA→ (TMP)-1→	→TMP →ALU
INX rp	0 0 R P	0 0 1 1				(RP) + 1 →	RP			
DCX rp	0 0 R P	1 0 1 1				(RP) - 1 →	RP			
DAD rp[8]	0 0 R P	1 0 0 1				X		(ri)→ACT	(L)→TMP, (ACT)+(TMP)→ALU	ALU→L, CY
DAA	0 0 1 0	0 1 1 1				DAA→A, FLAGS[10]				
ANA r	1 0 1 0	0 S S S	↓	↓	↓	(SSS)→TMP (A)→ACT		[9]	(ACT)+(TMP)→A	
ANA M	1 0 1 0	0 1 1 0	PC OUT STATUS	PC = PC + 1	INST→TMP/IR	(A)→ACT		HL OUT STATUS[6]	DATA→	→TMP

Table 3-4

CYCLE AND STATE TRANSITION SEQUENCES FOR EACH 8080 INSTRUCTION (Continued)

M3			M4			M5				
T1	T2[2]	T3	T1	T2[2]	T3	T1	T2[2]	T3	T4	T5
HL OUT STATUS[7]	(TMP) → DATA BUS									
PC OUT STATUS[6]	PC = PC + 1	B3 → rh								
	PC = PC + 1	B3 → W	WZ OUT STATUS[6]	DATA → A						
	PC = PC + 1	B3 → W	WZ OUT STATUS[7]	(A) → DATA BUS						
	PC = PC + 1	B3 → W	WZ OUT STATUS[6]	DATA → L WZ = WZ + 1		WZ OUT STATUS[6]	DATA → H			
PC OUT STATUS[6]	PC = PC + 1	B3 → W	WZ OUT STATUS[7]	(L) → DATA BUS WZ = WZ + 1		WZ OUT STATUS[7]	(H) → DATA BUS			
[9]	(ACT)+(TMP)→A									
[9]	(ACT)+(TMP)→A									
[9]	(ACT)+(TMP)+CY→A									
[9]	(ACT)+(TMP)+CY→A									
[9]	(ACT)-(TMP)→A									
[9]	(ACT)-(TMP)→A									
[9]	(ACT)-(TMP)-CY→A									
[9]	(ACT)-(TMP)-CY→A									
HL OUT STATUS[7]	ALU → DATA BUS									
HL OUT STATUS[7]	ALU → DATA BUS									
(rh)→ACT	(H)→TMP (ACT)+(TMP)+CY→ALU	ALU→H, CY								
[9]	(ACT)+(TMP)→A									

CYCLE AND STATE TRANSITION SEQUENCES FOR EACH 8080 INSTRUCTION (Continued)

MNEMONIC	OP CODE		M1[1]					M2		
	$D_7 D_6 D_5 D_4$	$D_3 D_2 D_1 D_0$	T1	T2[2]	T3	T4	T5	T1	T2[2]	T3
ANI data	1 1 1 0	0 1 1 0	PC OUT STATUS	PC = PC + 1	INST→TMP/IR	(A)→ACT		PC OUT STATUS[6]	PC = PC + 1	B2 → TMP
XRA r	1 0 1 0	1 S S S				(A)→ACT (SSS)→TMP		[9]	(ACT)+(TPM)→A	
XRA M	1 0 1 0	1 1 1 0				(A)→ACT		HL OUT STATUS[6]		DATA → TMP
XRI data	1 1 1 0	1 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1	B2 → TMP
ORA r	1 0 1 1	0 S S S				(A)→ACT (SSS)→TMP		[9]	(ACT)+(TMP)→A	
ORA M	1 0 1 1	0 1 1 0				(A)→ACT		HL OUT STATUS[6]		DATA → TMP
ORI data	1 1 1 1	0 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1	B2 → TMP
CMP r	1 0 1 1	1 S S S				(A)→ACT (SSS)→TMP		[9]	(ACT)-(TMP), FLAGS	
CMP M	1 0 1 1	1 1 1 0				(A)→ACT		HL OUT STATUS[6]		DATA → TMP
CPI data	1 1 1 1	1 1 1 0				(A)→ACT		PC OUT STATUS[6]	PC = PC + 1	B2 → TMP
RLC	0 0 0 0	0 1 1 1				(A)→ALU ROTATE		[9]	ALU→A, CY	
RRC	0 0 0 0	1 1 1 1				(A)→ALU ROTATE		[9]	ALU→A, CY	
RAL	0 0 0 1	0 1 1 1				(A), CY→ALU ROTATE		[9]	ALU→A, CY	
RAR	0 0 0 1	1 1 1 1				(A), CY→ALU ROTATE		[9]	ALU→A, CY	
CMA	0 0 1 0	1 1 1 1				$(\overline{A})$→A				
CMC	0 0 1 1	1 1 1 1				$\overline{CY}$→CY				
STC	0 0 1 1	0 1 1 1				1→CY				
JMP addr	1 1 0 0	0 0 1 1				X		PC OUT STATUS[6]	PC = PC + 1	B2 → Z
J cond addr[17]	1 1 C C	C 0 1 0				JUDGE CONDITION		PC OUT STATUS[6]	PC = PC + 1	B2 → Z
CALL addr	1 1 0 0	1 1 0 1				SP = SP - 1		PC OUT STATUS[6]	PC = PC + 1	B2 → Z
C cond addr[17]	1 1 C C	C 1 0 0				JUDGE CONDITION IF TRUE, SP = SP - 1		PC OUT STATUS[6]	PC = PC + 1	B2 → Z
RET	1 1 0 0	1 0 0 1				X		SP OUT STATUS[15]	SP = SP + 1	DATA → Z
R cond addr[17]	1 1 C C	C 0 0 0			INST→TMP/IR	JUDGE CONDITION[14]		SP OUT STATUS[15]	SP = SP + 1	DATA → Z
RST n	1 1 N N	N 1 1 1			ϕ→W INST→TMP/IR	SP = SP - 1		SP OUT STATUS[16]	SP = SP - 1	(PCH) → DATA BUS
PCHL	1 1 1 0	1 0 0 1			INST→TMP/IR	(HL) → PC				
PUSH rp	1 1 R P	0 1 0 1				SP = SP - 1		SP OUT STATUS[16]	SP = SP - 1	(rh) → DATA BUS
PUSH PSW	1 1 1 1	0 1 0 1				SP = SP - 1		SP OUT STATUS[16]	SP = SP - 1	(A) → DATA BUS
POP rp	1 1 R P	0 0 0 1				X		SP OUT STATUS[15]	SP = SP + 1	DATA → r1
POP PSW	1 1 1 1	0 0 0 1				X		SP OUT STATUS[15]	SP = SP + 1	DATA → FLAGS
XTHL	1 1 1 0	0 0 1 1				X		SP OUT STATUS[15]	SP = SP + 1	DATA → Z
IN port	1 1 0 1	1 0 1 1				X		PC OUT STATUS[6]	PC = PC + 1	B2 → Z, W
OUT port	1 1 0 1	0 0 1 1				X		PC OUT STATUS[6]	PC = PC + 1	B2 → Z, W
EI	1 1 1 1	1 0 1 1				SET INTE F/F				
DI	1 1 1 1	0 0 1 1				RESET INTE F/F				
HLT	0 1 1 1	0 1 1 0				X		PC OUT STATUS	HALT MODE[20]	
NOP	0 0 0 0	0 0 0 0	PC OUT STATUS	PC = PC + 1	INST→TMP/IR	X				

CYCLE AND STATE TRANSITION SEQUENCES FOR EACH 8080 INSTRUCTION (Continued)

M3			M4			M5						
T1	T2[2]	T3	T1	T2[2]	T3	T1	T2[2]	T3	T4	T5		
[9]	(ACT)+(TMP)→A											
[9]	(ACT)+(TMP)→A											
[9]	(ACT)+(TMP)→A											
[9]	(ACT)+(TMP)→A											
[9]	(ACT)+(TMP)→A											
[9]	(ACT)−(TMP); FLAGS											
[9]	(ACT)−(TMP); FLAGS											
PC OUT STATUS[6]	PC = PC + 1 B3 → W										WZ OUT STATUS[11]	(WZ) + 1 → PC
PC OUT STATUS[6]	PC = PC + 1 B3 → W										WZ OUT STATUS[11,12]	(WZ) + 1 → PC
PC OUT STATUS[6]	PC = PC + 1 B3 → W		SP OUT STATUS[16]	(PCH) → DATA BUS SP = SP − 1		SP OUT STATUS[16]	(PCL) → DATA BUS				WZ OUT STATUS[11]	(WZ) + 1 → PC
PC OUT STATUS[6]	PC = PC + 1 B3 → W[13]		SP OUT STATUS[16]	(PCH) → DATA BUS SP = SP − 1		SP OUT STATUS[16]	(PCL) → DATA BUS				WZ OUT STATUS[11,12]	(WZ) + 1 → PC
SP OUT STATUS[15]	SP = SP + 1 DATA → W										WZ OUT STATUS[11]	(WZ) + 1 → PC
SP OUT STATUS[15]	SP = SP + 1 DATA → W										WZ OUT STATUS[11,12]	(WZ) + 1 → PC
SP OUT STATUS[16]	(TMP = 00NNN000) → Z (PCL) → DATA BUS										WZ OUT STATUS[11]	(WZ) + 1 → PC
SP OUT STATUS[16]	(rl) → DATA BUS											
SP OUT STATUS[16]	FLAGS → DATA BUS											
SP OUT STATUS[15]	SP = SP + 1 DATA → rh											
SP OUT STATUS[15]	SP = SP + 1 DATA → A											
SP OUT STATUS[15]	DATA → W		SP OUT STATUS[16]	(H) → DATA BUS		SP OUT STATUS[16]	(L) → DATA BUS	(WZ) → HL				
WZ OUT STATUS[18]	DATA → A											
WZ OUT STATUS[18]	(A) → DATA BUS											

NOTES:

1. The first memory cycle (M1) is always an instruction fetch; the first (or only) byte, containing the op code, is fetched during this cycle.

2. If the READY input from memory is not high during T2 of each memory cycle, the processor will enter a wait state (TW) until READY is sampled as high.

3. States T4 and T5 are present, as required, for operations which are completely internal to the CPU. The contents of the internal bus during T4 and T5 are available at the data bus; this is designed for testing purposes only. An "X" denotes that the state is present, but is only used for such internal operations as instruction decoding.

4. Only register pairs rp = B (registers B and C) or rp = D (registers D and E) may be specified.

5. These states are skipped.

6. Memory read sub-cycles; an instruction or data word will be read.

7. Memory write sub-cycle.

8. The READY signal is not required during the second and third sub-cycles (M2 and M3). The HOLD signal is accepted during M2 and M3. The SYNC signal is not generated during M2 and M3. During the execution of DAD, M2 and M3 are required for an internal register-pair add; memory is not referenced.

9. The results of these arithmetic, logical or rotate instructions are not moved into the accumulator (A) until state T2 of the next instruction cycle. That is, A is loaded while the next instruction is being fetched; this overlapping of operations allows for faster processing.

10. If the value of the least significant 4-bits of the accumulator is greater than 9 or if the auxiliary carry bit is set, 6 is added to the accumulator. If the value of the most significant 4-bits of the accumulator is now greater than 9, or if the carry bit is set, 6 is added to the most significant 4-bits of the accumulator.

11. This represents the first sub-cycle (the instruction fetch) of the next instruction cycle.

12. If the condition was met, the contents of the register pair WZ are output on the address lines (A_{0-15}) instead of the contents of the program counter (PC).

13. If the condition was not met, sub-cycles M4 and M5 are skipped; the processor instead proceeds immediately to the instruction fetch (M1) of the next instruction cycle.

14. If the condition was not met, sub-cycles M2 and M3 are skipped; the processor instead proceeds immediately to the instruction fetch (M1) of the next instruction cycle.

15. Stack read sub-cycle.

16. Stack write sub-cycle.

17.

CONDITION	CCC
NZ – not zero (Z = 0)	000
Z – zero (Z = 1)	001
NC – no carry (CY = 0)	010
C – carry (CY = 1)	011
PO – parity odd (P = 0)	100
PE – parity even (P = 1)	101
P – plus (S = 0)	110
M – minus (S = 1)	111

18. I/O sub-cycle: the I/O port's 8-bit select code is duplicated on address lines 0-7 (A_{0-7}) and 8-15 (A_{8-15}).

19. Output sub-cycle.

20. The processor will remain idle in the halt state until an interrupt, a reset or a hold is accepted. When a hold request is accepted, the CPU enters the hold mode; after the hold mode is terminated, the processor returns to the halt state. After a reset is accepted, the processor begins execution at memory location zero. After an interrupt is accepted, the processor executes the instruction forced onto the data bus (usually a restart instruction).

SSS or DDD	Value	rp	Value
A	111	B	00
B	000	D	01
C	001	H	10
D	010	SP	11
E	011		
H	100		
L	101		

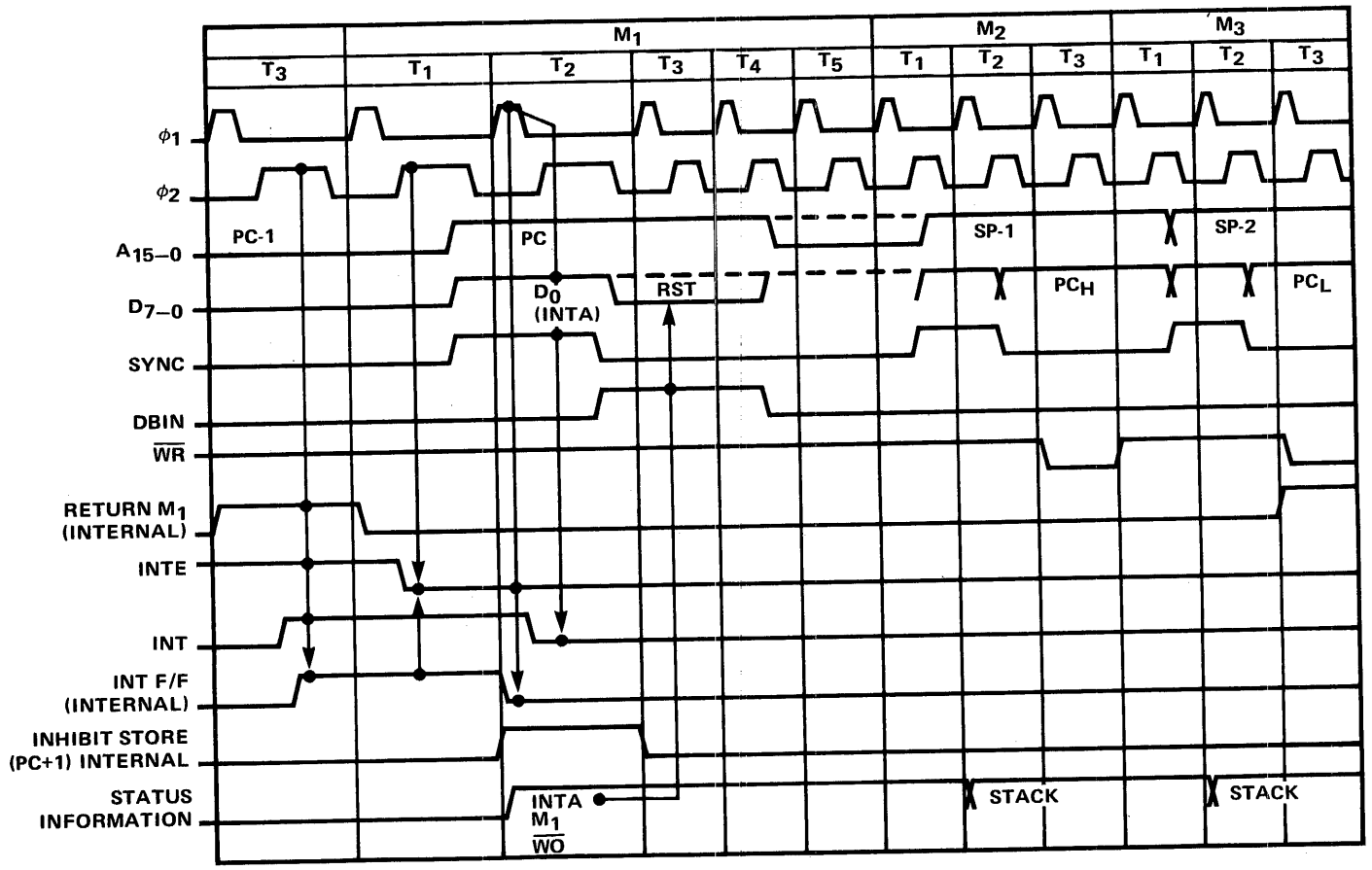


Figure 3-9. Interrupt Timing

3.3.4 HOLD SEQUENCES

By applying a HOLD to the appropriate control pin on the processor, an external device can cause the CPU to suspend its normal operations and relinquish control of the address and data busses. The processor responds to a request of this kind of floating its address and data outputs, so that these exhibit a high impedance to other devices sharing the busses. At the same time, the processor acknowledges the HOLD by placing a high on its HLDA output pin. During an acknowledged HOLD, the address and data busses are under control of the peripheral which originated the request, enabling it to conduct memory transfers without processor intervention. The HOLD provision, however, is not used on the Central Processor Module.

3.3.5 HALT SEQUENCES

When a halt instruction (HLT) is executed, the CPU enters the halt state (T_{WH}) after state T2 of the next machine cycle. There are only three ways in which the 8080 can exit the halt state:

- A high on the RESET line will always reset the 8080 to state T1; RESET also clears the program counter.
- A HOLD input will cause the 8080 to enter the hold state, as previously described. When the HOLD lines goes low, the 8080 re-enters the halt state on the rising edge of the next ϕ_1 clock pulse
- An interrupt (i.e., INT goes high while INTE is enabled) will cause the 8080 to exit the halt state and enter state T1 on the rising edge of the next ϕ_1 clock pulse.

NOTE: The interrupt enable (INTE) flag must be set when the halt state is entered; otherwise, the 8080 will only be able to exit via a RESET signal.

3.3.6 START-UP OF THE 8080 CPU

When power is applied initially to the 8080, the processor begins operating immediately. The contents of its program counter, stack pointer, and the other working registers are naturally subject to random factors and cannot be specified. For this reason, it will be necessary to begin the power-up sequence with RESET.

An external RESET signal of three clock period duration (minimum) restores the processor's internal program counter to zero. Program execution thus begins with memory location zero, following a RESET. Systems which require the processor to wait for an explicit start-up signal will store a halt instruction (HLT) in this location. A manual or an automatic INTERRUPT will be used for starting. In other systems, the processor may begin executing its stored program immediately. Note, however, that the RESET has no effect on status flags, or on any of the processor's working registers (accumulator, indices, or stack pointer). The contents of these registers remain indeterminate, until initialized explicitly by the program.

3.4 PERIPHERAL LOGIC: THEORY OF OPERATION

In this section, we describe the peripheral logic on the Central Processor Module, the logic which directly supports the activities of the 8080 CPU. We begin by explaining the clock generator logic, since all the operations of the module are ultimately referred to signals generated in this section. We then describe the bus control logic, which resolves all exchanges of bus control between the CPU Module and other modules capable of controlling the bus (i.e., other "master" modules). Finally, we give descriptive examples of all module operations, showing how the peripheral logic extends the basic capabilities of the 8080 processor.

The schematic for the Central Processor Module is provided in Figure 3-19, located in Section 3.4.8. You will probably find it helpful to refer to this schematic as you read the following sections.

3.4.1 CLOCK GENERATOR LOGIC

The clock generator logic consists of a crystal-controlled clock oscillator, a counter, level shifting provisions, and miscellaneous counting and gating circuits. These are shown on sheet 4 of the module schematic, Figure 3-19.

The clock oscillator furnishes a 32-MHz signal to the input of the counting section, which uses it to develop the ϕ_1 and ϕ_2 clock signals used to generate the remaining timing outputs.

A 32-MHz quartz crystal, operating in the series-resonant mode, is the basic frequency reference. The crystal acts as a bandpass filter at the desired frequency. It thus permits a portion of the signal developed across the capacitive divider in the translator's collector circuit to reach the emitter, in proper phase to sustain oscillation. The output from the oscillator state is coupled to a second state, biased to operate as an over-driven amplifier, and the shaped output of the second is used to drive the synchronous counter chain.

Four 74S114 high-speed J-K flip-flops constitute the clock counter. This is a synchronous configuration, with the steering function obtained through the use of external coincidence gates. A slight variation on conventional practice produces a fourth stage output which is "displaced" with respect to the outputs of the first three stages, by one full period of the driving clock. In all other respects, however, the counter resembles the familiar modulo-16 synchronous counters in common Idealized waveforms are shown in Figure 3-10.

The 2-MHz output of the fourth counting stage becomes the ϕ_2 clock signal. Coincidence in the outputs of the third and fourth stages generates the ϕ_1 clock. As Figure 3-10 shows, this produces two non-overlapping clock signals, with characteristic pulse widths of 125 and 250 ns and separation intervals of approximately 31 and 94 ns.

The ϕ_1 and ϕ_2 clock phases are applied to the inputs of an MH0026 level shifter; the shifter outputs are then used to drive the 8080's MOS-level clock inputs (CPU pins 22 and 15).

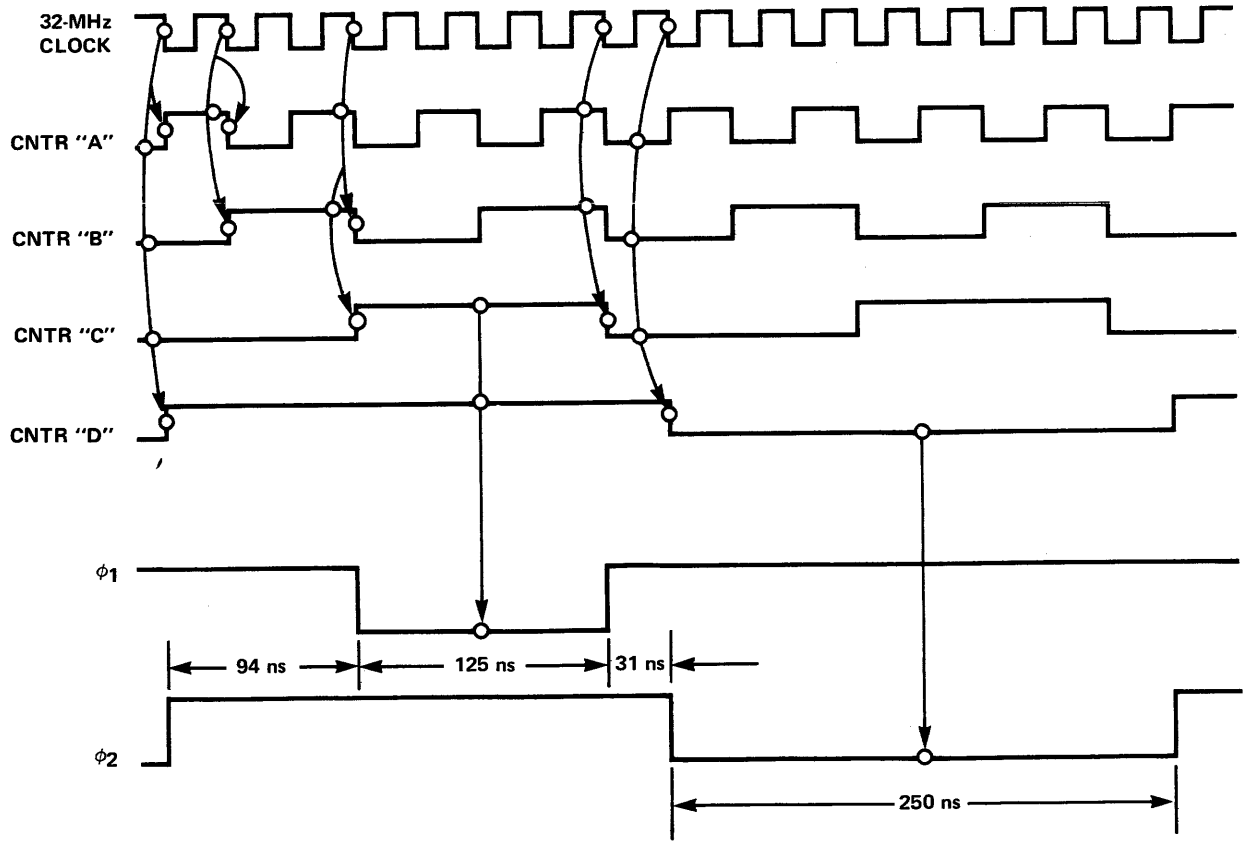


Figure 3-10. Oscillator–Counter Timing

3.4.2 BUS CONTROL LOGIC

The Central Processor Module's bus control logic consists of two J-K flip-flops and miscellaneous gating circuits. These are shown on sheet 2 of the module schematic, Figure 3-19.

Bus contention is resolved on each occurrence of the bus clock (BCLK) signal. BCLK/ is received at bus pin 13, inverted and applied to the clock inputs of the two 74109 J-K flip-flops (A20-12 and A20-4), and one of the 74H74 D-type flip-flops (A21-11).

The J and $\overline{K}$ inputs to the first flip-flop are wired to the halt status inverter (A32-8) and to pin 29 of the auxiliary connector (J2). In configurations with more than one "bus master" module, an external request (XREQST/) signal can be input through J2-29. XREQST/ indicates that another module requests use of the bus. If XREQST/ is present and active (low), the flip-flop will reset with the occurrence of BCLK. Otherwise, the flip-flop sets. The $\overline{Q}$ output is inverted and driven through bus pin 18 as BREQ/. When active, BREQ/ indicates that the CPU module requests use of the bus. The bus request flip-flop is pre-reset by the initialization (INIT/) signal.

When low, the $\overline{Q}$ output of the second J-K flip-flop enables an 8098 tri-state inverter to drive the bus busy (BUSY/) signal (bus pin 17). When true, BUSY/ indicates that the CPU module or another master module has control of the bus. If BUSY/ is not already active, the busy flip-flop will reset and activate BUSY/ on the occurrence of BCLK, when the following conditions are true:

(1) The bus request flip-flop is set *and*,

(2) The bus priority in (BPRN/) signal is true, BPRN/ (bus pin 15) indicates that no higher bus priority module is requesting the bus.

The busy flip-flop is pre-set by the initialization (INIT/) signal.

When the CPU module has control of the bus, the $\overline{Q}$ output of the busy flip-flop (A20-7) enables the selected (SEL) signal. In addition, this $\overline{Q}$ output serves as the D input to a 74H74 section (A21-2). If BUSY/ is true, this D-type flip-flop sets with the occurrence of BCLK. Unless it is set, the command enable (CMDE/) signal cannot be generated. This provides a one cycle delay between address and command outputs, during the CPU module recovery of the bus.

It is important that the CPU module not lose control of the bus while in the middle of a transfer. Three D-type flip-flops [one 7474 (A10) and two 74H74's (A8)] synchronize transitions of the BUSY/ and SEL signals with the beginning of a machine cycle (i.e., before transfers are initiated). The first flip-flop sets on the negative-going edge of ϕ_2 if SYNC is true (i.e., during the first state, T1, of a machine cycle). The output of this section, in turn, clocks the second flip-flop, the output of which is gated through to the J input of the busy flip-flop and the NAND gate that drives SEL. If the CPU module loses control of the bus, this second flip-flop resets. When the CPU module regains control of the bus, the third flip-flop is clocked by SEL and, in turn, pre-sets the second flip-flop, thus completing recovery of the bus control logic.

Whenever the CPU module relinquishes control of the bus (SEL goes false), the 8080 CPU enters a wait state. The absence of SEL, which feeds the D input of one of the ready flip-flops (A7-12), prevents READY from being generated. The lack of an active READY input forces the 8080 into the wait state. When the bus is regained, READY is again enabled.

If it is necessary to guarantee that one or more multi-byte transfers not be interrupted by loss of the bus, an override function can be invoked by the program (refer to Section 3.4.7). Execution of an output instruction to address FE_{16} controls the override function. If data bit 0 (D0) is high when the output to FE_{16} instruction is executed, the override flip-flop (a 7474 section) is set; OVERRIDE/ goes true. OVERRIDE/ is gated through to the J input of the busy flip-flop and the D input of the second D-type section (A8-2). While OVERRIDE/ is true, the bus control logic is prevented from relinquishing control of the bus. If data bit 0 is low when an output to FE_{16} is executed or if the initialization (INIT/) signal occurs, the override flip-flop resets; OVERRIDE/ goes false.

The bus control logic is primarily controlled by the state of the bus priority in (BPRN/) signal. When BPRN/ is true, the module can gain or retain control of the bus, and when BPRN/ is false, the module will relinquish control of the bus, unless override has been invoked. BPRN/ may be generated by a central *parallel* priority network; in the INTELLEC MDS System such a network is included on the Front Panel Control Module. BPRN/ may also be generated and transmitted in *serial*. BPRN/ is captured by the highest priority module requiring control of the bus. Those modules that do not require the bus accept BPRN/ and pass BPRO/ (bus priority out) on to the next module on the bus. Thus, a module's priority is dependent on its relative position on the bus. To use the CPU module in a general serial priority scheme, jumper pad 1-2 must be connected, tying BPRN/ to BPRO/ (bus pin 16).

Timine for the bus control logic is shown in Figure 3-11.

NOTE: Halt and interrupt cycles affect the bus control logic in a unique manner. If the 8080 CPU enters the halt state, the HLTA status line at flip-flop A31 is gated to the J-K inputs of the transfer request flip-flop (A20), causing removal of the XREQST signal. Consequently, control of the bus is relinquished. Recovery from this state is achieved *only* by interruption of the 8080 CPU. The halt status gate (A3-6) allows the 8080 to receive a ready input for processing of the interrupt instruction, regardless of bus control status. The CPU then makes a transfer request and bus operations are restarted after bus recovery.

3.4.3 INSTRUCTION FETCH AND MEMORY READ

An instruction fetch and any other memory read cycle appear the same to the peripheral logic. Only

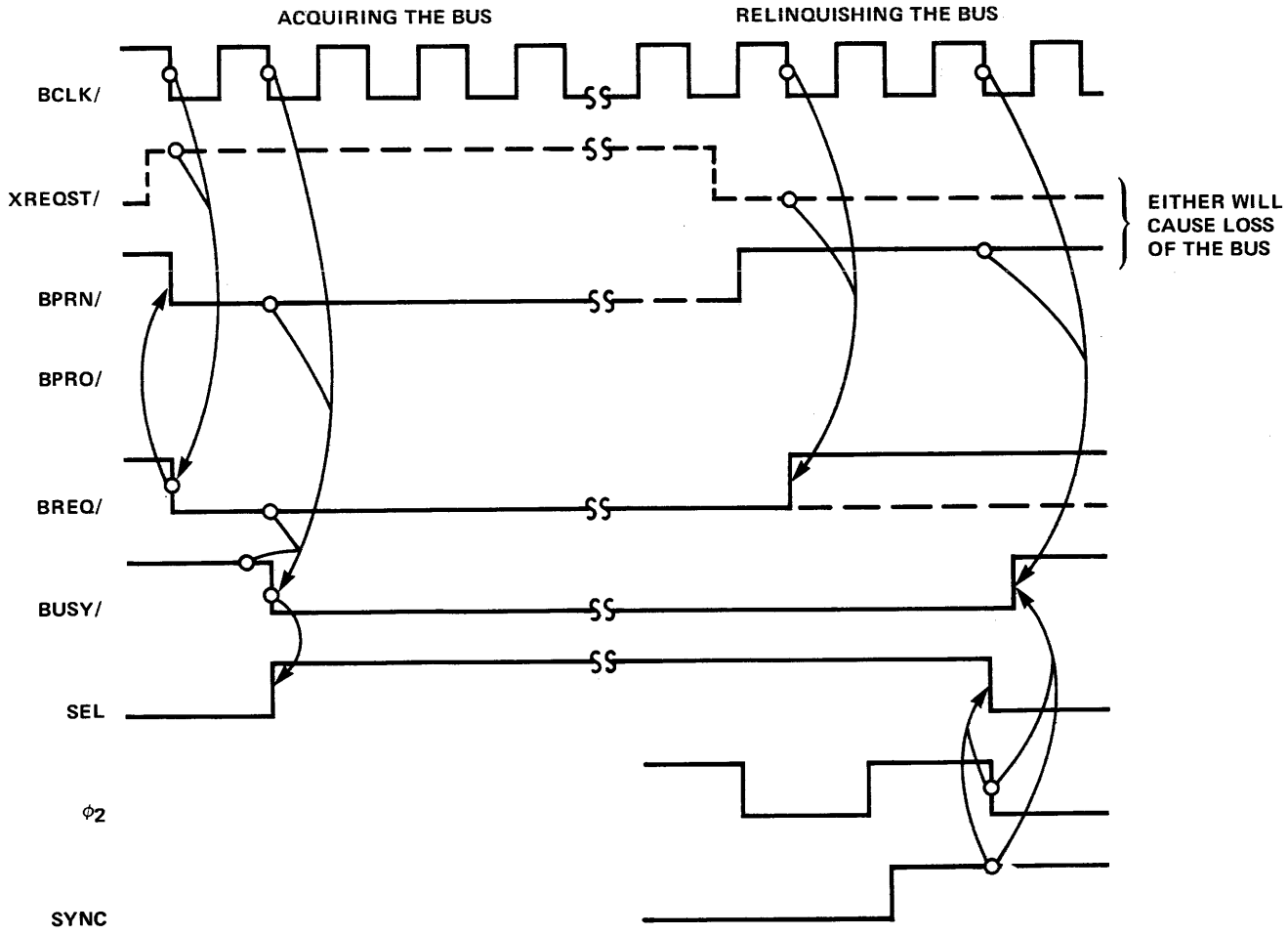


Figure 3-11. Bus Control Logic Timing

the 8080 distinguishes between the two. Status bit 7 (MEMR) is generated by the 8080 for both types of cycles. While status bit 5 (M1) is only generated during an instruction fetch, the peripheral logic does not examine this bit, so it is not aware of the difference. Consequently, the following description of how the peripheral logic responds to a memory read, also applies to an instruction fetch. Refer to Section 3.3.2 for a complete discussion of how the 8080 CPU distinguishes between the two types of cycles.

During state T1, the 8080 processor transmits a 16-bit address on its lines (A0–A15). Assuming that the CPU module has control of the system bus (i.e., if SEL is true), the 16-bit address passes through the 16 tri-state inverters (which are enabled by ADEN/) and is presented to the external memory. Information placed on the address bus remains stable until the T4 internal processing state.

Status information is also broadcast during the T1 state, on the processor's eight data lines (D0–D7). The eight status bits are buffered by an 8212 device and passed to the CPU module's command generation logic (see sheet 2 of the module schematic, Figure 3-19). During any memory read cycle (i.e., instruction fetch, stack read or memory read), the MEMR status bit (D7) is true. The status information remains stable for one clock period, during which SYNC is also active.

If MEMR (D7) is true and the HLTA (D3) status bit is false (i.e., the 8080 is not halted), the memory read latch in the command generation logic sets on the rising-edge of the first ϕ_1 pulse after SYNC is issued. The $\overline{Q}$ output (MRDC/) is made available (from pin 19) to external memory via an 8095 non-inverting driver. MRDC/ is the memory read command.

The processor issues DBIN in the latter portion of state T2; DBIN remains stable until the latter portion of state T3, even if one or more wait states intervene between T2 and T3.

When memory responds to the MRDC/ command by placing the addressed data byte on the bus, it issues an external acknowledge (XACK/) signal which is received at pin 23 on the CPU module. XACK/ enables the generation of READY on the rising edge of the next ϕ_2 pulse. The 8080 processor will advance to state T3 only after READY goes true. If XACK/ is not returned prior to T2-ϕ_2, one or more wait states will occur between T2 and T3.

A special provision has been implemented that allows the 16K RAM Module to be accessed without necessitating a wait state, even though the RAM Module is not capable of returning XACK/ by T2-ϕ_2. Because the RAM Module is fast enough to have stable data on the bus by the beginning of state T3 (as required), the RAM Module has been designed to generate an advanced acknowledge signal (AACK/), that anticipates having the data ready in time. The presence of AACK/ (pin 25) allows READY to be generated early enough in T2 to prevent the occurrence of a wait state, thus greatly increasing the efficiency of the RAM Module.

DBIN gates the data byte (on lines DAT$\emptyset$/–DAT7/) through two 8226 parallel bidirectional bus drivers and into the 8080. The trailing edge of DBIN resets the memory read (MRDC/) latch.

Memory read timing is shown in Figure 3-12. The diagram illustrates the use of XACK/ with a single wait state, as well as 8080/ without a wait state. READY is generated by the first acknowledgement to appear (also refer to NOTE at the end of Section 3.4.4).

3.4.4 MEMORY WRITE

A memory write cycle proceeds in much the same fashion as a memory read. During state T1, the 8080 processor transmits a 16-bit address. Assuming that the CPU module has control of the system bus, the 16-bit address is presented to external memory via 16 enabled tri-state inverters.

Status information is broadcast during state T1, over the eight data lines. During memory write cycles, all status bits are low. The low levels on data lines 1 and 4 (status bits $\overline{WO}$ and OUT, respectively) are gated through to the D input of the memory write latch causing the latch to set on the rising edge of the first ϕ_1 pulse after SYNC is issued. The $\overline{Q}$ output (MTWC/) is made available

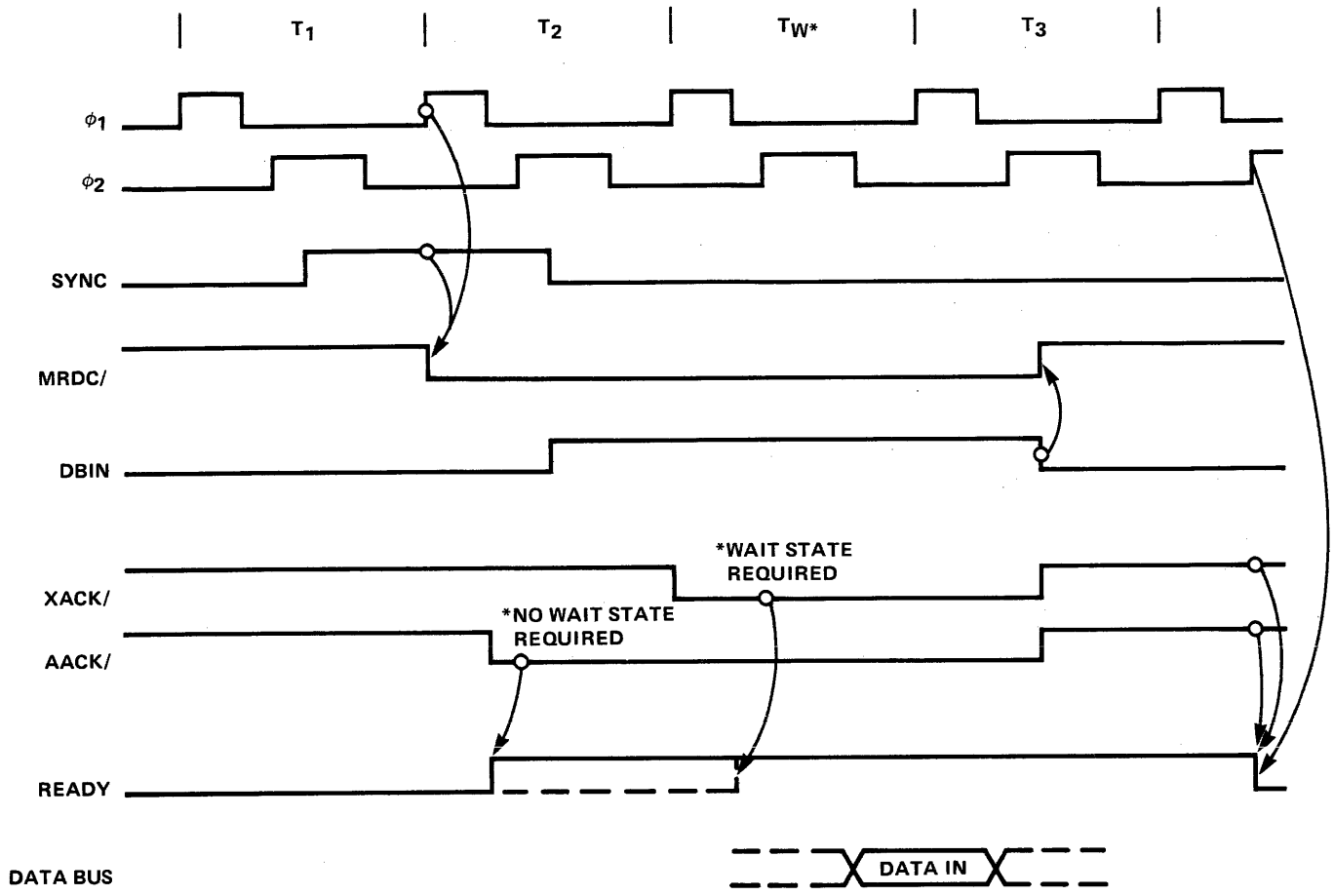


Figure 3-12. Memory Read Timing

(from pin 20) to external memory via an 8095 non-inverting driver. MTWC/ is the memory write command.

At the end of state T2, the processor places the data byte on the data lines and, subsequently, issues WR/. The data byte and WR/ remain stable through state T3 and any intervening wait states. The absence of DBIN enables the data byte through the two 8226 parallel, bidirectional bus drivers. The absence or presence of DBIN dictates direction for the 8226 bidirectional drivers during all data bus transfers.

When the memory device accepts the data, it issues XACK/ which enables the generation of READY on the CPU module. If XACK/ is not received prior to the rising edge of ϕ_2 during state T2, one or more wait states will be required before the processor can advance to state T3.

As we mentioned in the previous section, memory references to the RAM Module proceed somewhat differently. To fully utilize the RAM's fast access time, the module anticipates accepting data and issues an advanced acknowledgement (AACK/) which eliminates the need for any wait states. The AACK/ signal (pin 25) enables READY during state T2.

The trailing edge of the write strobe (WR/) resets the memory write (MWTC/) latch.

Memory write timing is shown in Figure 3-13.

NOTE: The timing for memory read and write cycles (Figures 3-12 and 3-13, respectively) is compatible with the INTELLEC MDS Bus specifications, with two exceptions. These exceptions are included to allow the 8080 processor to run at full speed for memory read and write operations. Neither of the exceptions severly limit the flexibility of the

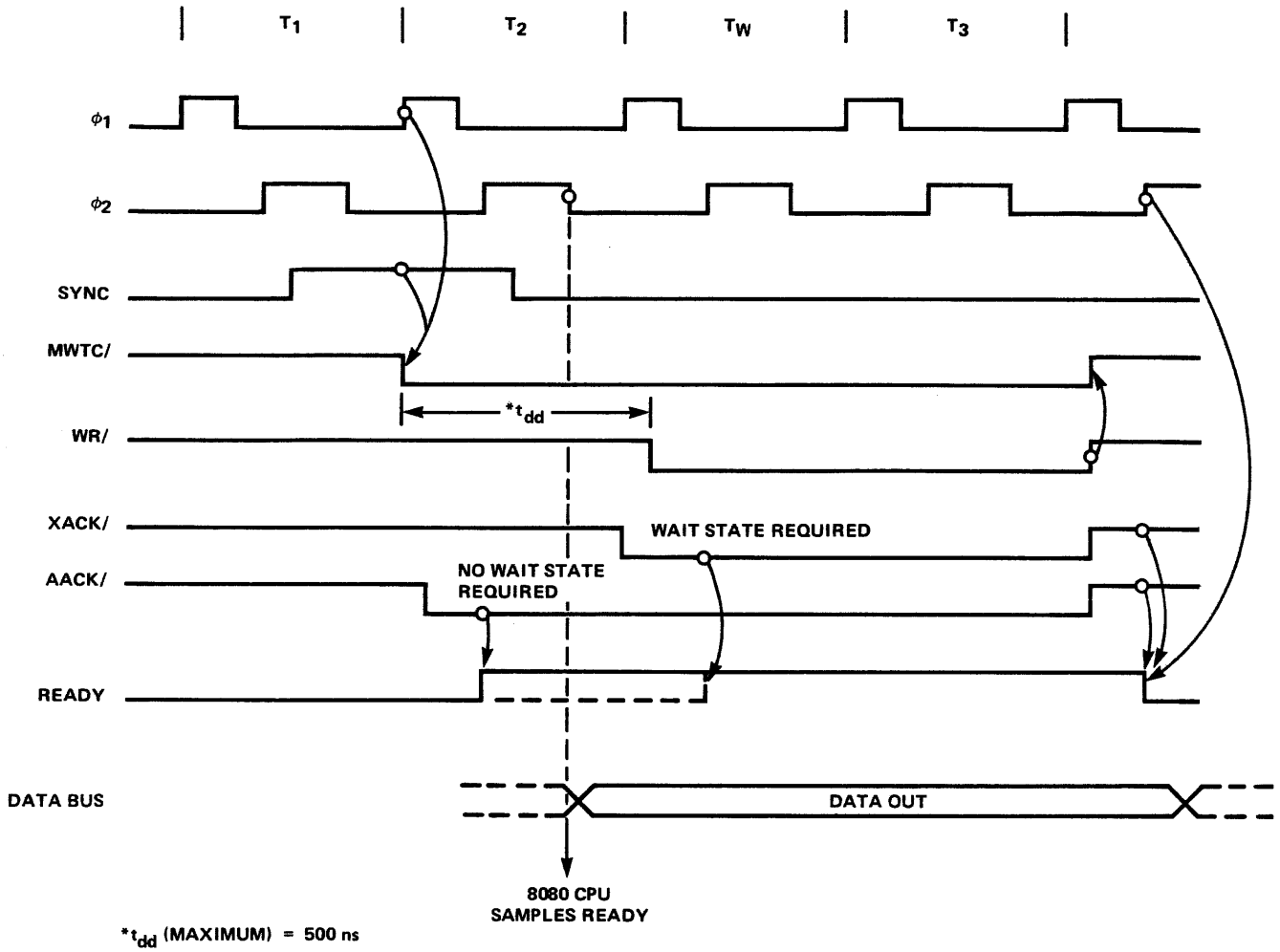


Figure 3-13. Memory Write Timing

module. In fact, they may be optionally overridden* to achieve full compatibility with the bus if speed is not critical.

The exceptions cited above are the use of an advanced acknowledge (AACK/) and the delay to stable data during a memory write operation (t_{DD}). Both stem from the manner in which the READY input is sampled by the 8080 CPU. The MDS Bus specification requires that an acknowledge be returned only if the "slave" device has both received a command and completed the necessary data transfer. The 8080 CPU, however, samples its READY input prior to generating the leading edge of its two basic commands, DBIN for read and WR/ for write. If DBIN and WR/ are used as memory commands, the first 8080 READY sampling will find an MDS slave device "not ready" since the command has yet to reach the bus (see Figure 3-13). The consequence would be an unavoidable 500-ns delay, regardless of the "slave" device's response time. This may result in up to a 33% and 14% decrease in the operating speeds of memory read and write operations, respectively.

To avoid this degradation, commands are initiated earlier in the cycle (i.e., MRDC/ and MWTC/ precede DBIN and WR/, respectively), and the advanced acknowledge (AACK/) is allowed. This permits a "slave" device to return a ready condition earlier than the 8080 CPU's sampling point.

*The jumper labeled "advanced write" must be moved from a D-C connection to an E-D connection and the AACK/ line must be disabled on the CPU or memory modules.

3.4.5 INPUT/OUTPUT

Input and output cycles, like other types of machine cycles, are identified by a unique combination of status bits that appear on the eight data lines during state T1, coincidentally with SYNC. Input cycles are indicated by a high level on data line 6 (INP); output cycles by a high level on data line 3 (OUT).

INP is applied to the D input of the I/O read latch. During input cycles, the latch sets on the rising edge of the first ϕ_1 pulse after SYNC. The $\overline{Q}$ output (IORC/) is made available (from pin 21) to all external devices via an 8095 non-inverting device. IORC/ is the I/O read command.

As we described in the previous section, the OUT and $\overline{WO}$ status bits are applied to the inverted inputs of a 74S02 gate that feeds the D input of the memory write latch. During memory write cycles the low levels on OUT and $\overline{WO}$ enable the latch to set.

During output (I/O write) cycles, however, the high level on OUT prevents the latch from setting. The high $\overline{Q}$ output, instead, feeds a 74H00 NAND gate. On the rising edge of the write strobe (WR), the gate is activated and the resulting output (IOWC/) is made available (from pin 22) to all external devices via an 8095 non-inverting driver. IOWC/ is the I/O write command. Because IOWC/ is dependent on the presence of WR (which does not occur until after state T2), all output cycles will incur at least one wait state regardless of device speeds.

The I/O device to be accessed is identified by an 8-bit address that is duplicated on address lines A0–A7 and A8–A15 during state T1. Assuming that the CPU module has control of the system bus (i.e., if SEL is true), the 16 address lines are enabled through 16 tri-state inverters and presented to all external devices. The address lines remain stable until state T4.

During input cycles, the 8080 issues DBIN in the latter portion of state T2; DBIN remains stable until the latter portion of state T3, even if one or more wait states intervene. DBIN is subsequently used to strobethe input data into the 8080.

During output cycles, the processor issues WR/ at the beginning of the first wait state. WR/ remains stable through state T3 and all of the wait states that precede it. Just prior to generating WR/, the processor places a data byte on the data lines. The absence of DBIN during output cycles enables the data byte through the two 8226 parallel, bidirectional bus drivers and out onto the system bus.

When the addressed I/O device responds to the IORC/ or IOWC/ command by inputting a data byte on the data bus or by accepting the data output by the processor, the I/O device returns an acknowledgement (XACK/) signal to the CPU module.

XACK/ enables READY which, in turn, allows the processor to proceed with state T3.

During output cycles, the processor maintains stable levels on the WR/ and data lines through state T3. IOWC/ goes inactive with the trailing edge of WR.

During state T3 of a input cycle, DBIN gates the data byte through the two 8226 parallel, bidirectional bus drivers and into the 8080 processor. The trailing edge of DBIN resets the I/O read (IORC/) flip-flop.

I/O timing is illustrated in Figure 3-14.

3.4.6 INTERRUPTS

The interrupt logic for the Central Processor Module is shown on sheet 3 of the module schematic, Figure 3-19.

A device requests an interrupt by pulling one of the eight interrupt level request lines (INTØ/–INT7/) low. The request is applied to one of the inverted inputs of a 7432 negative-NAND gate. The other gate input is furnished by one of the outputs of an 8212 latch/buffer. This 8212 stores the program-controlled interrupt mask (see Section 3.4.7). Unless the interrupting level has been masked out (i.e., disabled), the interrupt request is gated through the 7432 section and applied to the appropriate priority request input of a 3214 interrupt control unit. The 3214 latches the request(s), resolves priority among simultaneous requests, and

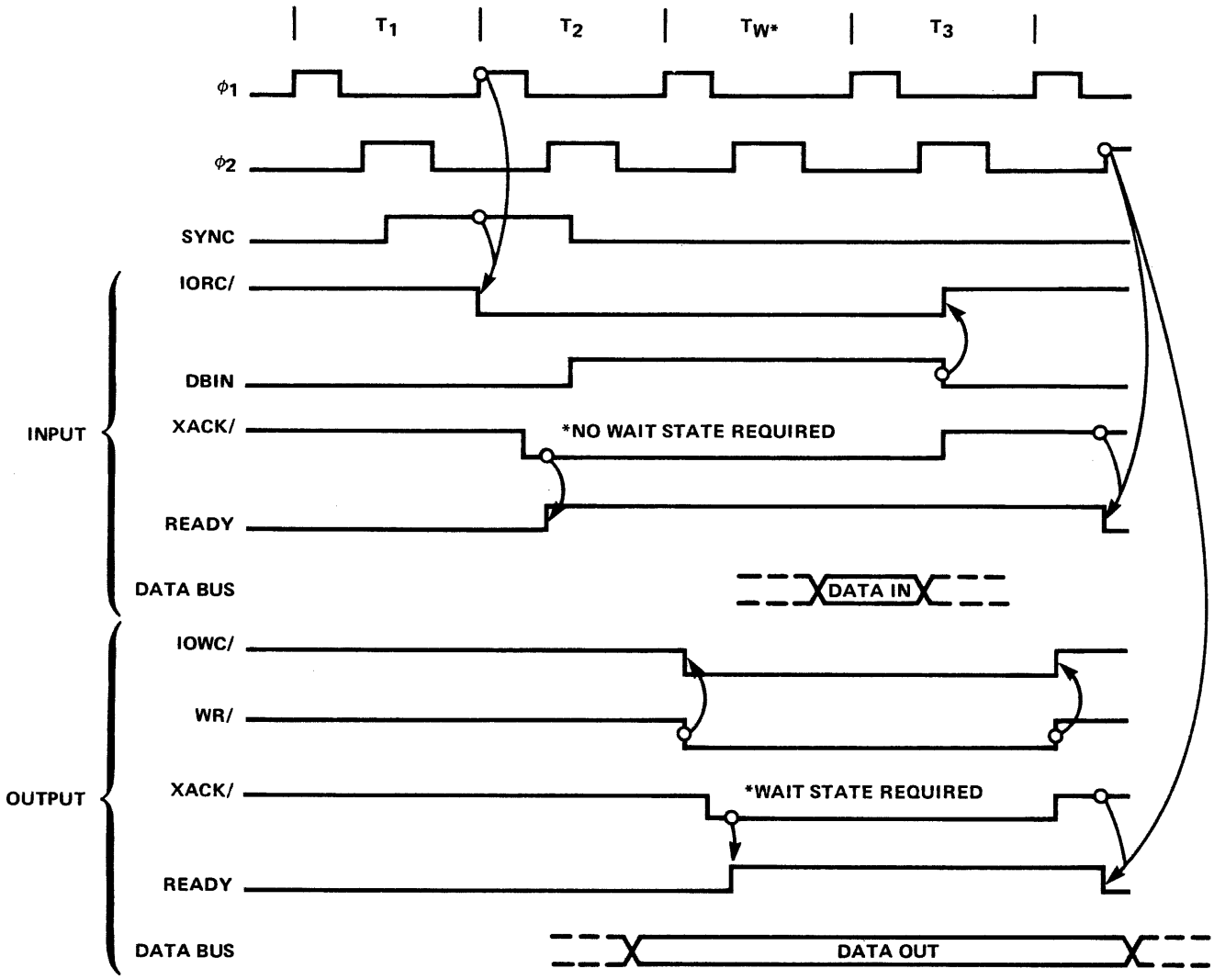


Figure 3-14. I/O Timing

issues an inverted 3-bit binary-encoded output (A0–A2) that reflects the highest priority requesting level. The three encoded bits (INTV3–INVT5) are applied to one of the 8212 latch/buffers that drive the data bus into the processor. The level indicator bits are also available to external modules via auxiliary connector pins P2-52, 50 and 48. In addition, the three bits output by the 3214 are inverted and applied to the data inputs of a 3101 RAM element. The 3214 also uses the 3-bit level indicator internally (see Figure 3-15). The three bits are fed into the 3214's priority comparator where they are compared with the interrupt level currently being serviced. If the requesting level is of higher priority than the current level (as indicated by the output of the 3214's current status latch), the 3214's INT ACK FF sets on the leading edge of ϕ_2. The INT ACK FF output is inverted (IMB) and applied to the pre-set input of a high-speed D-type latch (A11-4). The output of this latch (INT) is fed directly into the 8080 processor's interrupt input. The INT ACK FF output also pre-sets the INT DIS FF within the 3214, thus preventing any new interrupts until after the 3214's current status latch has been updated. If the requesting level had lower priority than the current level, the new request would have been ignored. Table 3-5 lists relative interrupt level priorities.

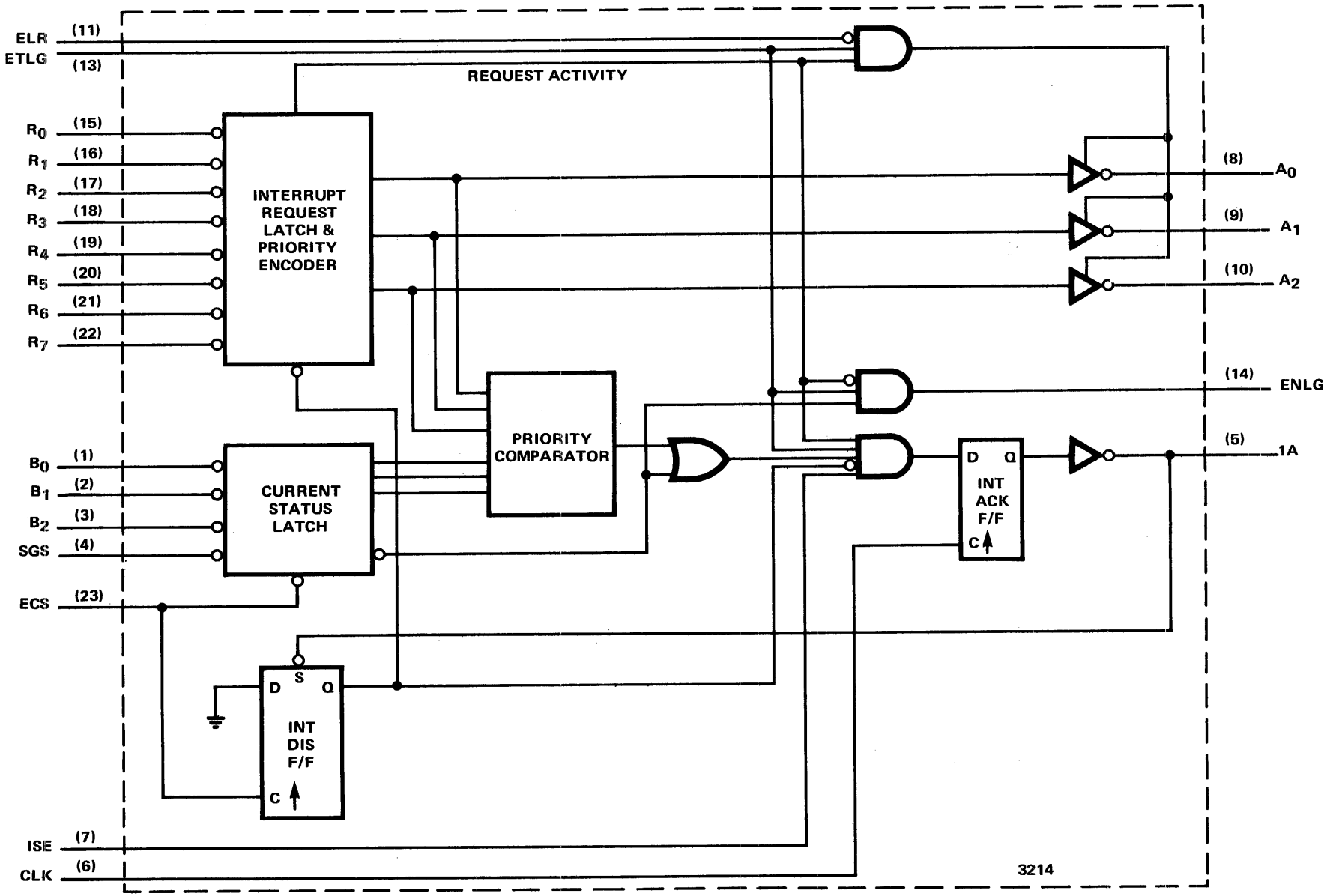


Figure 3-15. 3214 Block Diagram

Table 3-5

INTERRUPT LEVEL PRIORITIES

EXTERNAL INTERRUPT LEVEL			PRIORITY	RESTART INSTRUCTION	CALLED ADDRESS	INTERRUPT MASK BIT (1=DISABLE)
#	MNEMONIC	PIN				
Ø	INTØ/	41	Highest	Ø	ØØØØ	Bit Ø
1	INT1/	42	2	1	ØØØ8	Bit 1
2	INT2/	39	3	2	ØØ1Ø	Bit 2
3	INT3/	40	4	3	ØØ18	Bit 3
4	INT4/	37	5	4	ØØ2Ø	Bit 4
5	INT5/	38	6	5	ØØ28	Bit 5
6	INT6/	35	7	6	ØØ3Ø	Bit 6
7	INT7/	36	Lowest	7	ØØ38	Bit 7

After completing the machine cycle in progress, the processor acknowledges the interrupt. This it does by entering an alternative interrupt machine cycle, instead of proceeding directly to the next instruction fetch. As we explained in Section 3.3, the processor does not increment the internal program counter as it normally would. Consequently, the logic sequence of the interrupted program is maintained. When the interrupt has been serviced, the interrupted program can be resumed with no loss of continuity.

During state T1 of an interrupt machine cycle, the processor issues an INTA status bit over data line 0. INTA uniquely identifies the cycle as an interrupt machine cycle.

INTA (DØ) is applied to the D input of a D-type latch within the command generation logic (sheet 2 of the module schematic). The simultaneous occurrence of SYNC and ϕ_2 clocks the latch set. The resulting output (INTA/) generates local acknowledgement (LACK/) and disable bus (DISB/) signals and increments a 74191 counter whose four outputs serve as the address for the 3101 RAM element. LACK/ enables the READY flip-flop. In the event that the CPU module does not have control of the bus when an interrupt occurs, the output of gate A3-6 enables READY (A7-8) for the interrupt cycle only. READY, in turn, allows the 8080 processor to proceed from state T2 to state T3. LACK/ is necessary because no external device returns an external acknowledgement (XACK/ or AACK/) during interrupt cycles. DISB/ disables the two 8226 bidirectional bus drivers so that random data on the external bus does not interfere with the Restart instruction (RST) that is forced onto the 8080 processor data lines (by an 8212 I/O port) during state T3.

At the end of state T2, the processor issues DBIN in anticipation of accepting the Restart (RST n) instruction. Recall that the three level indicators, INV3–INV5, were applied to the inputs of an 8212 I/O port. The other five inputs are tied to +5 volts. Consequently, the eight inputs to this 8212 device constitute the machine code for a RST n instruction (11 NNN111), where NNN are the three encoded interrupt vector bits. The simultaneous occurrence of DBIN and INTA/ enables the 8212 section, which, in turn, furnishes the processor with the RST n instruction. The processor branches to the instruction whose address is eight times the value of NNN (see Table 3-5).

The presence of DBIN and INTA/ also allows for updating of the current status latch within the 3214 interrupt control unit and the nested priority table that is stored in the 3101 RAM element. Recall that the three interrupt level indicators were applied to the data inputs of the 3101 RAM device and that the address inputs to the 3101 (from the

74191 counter) were incremented with the issuance of INTA/ at the beginning of the interrupt machine cycle. When DBIN appears, it, in conjunction with INTA/, strobes the write enable input to the 3101 RAM, causing the interrupt level indicator to be written into the nested priority table. This level indicator value subsequently appears on the RAM output lines, which are applied to the current status latch inputs of the 3214 interrupt control unit. DBIN and INTA/ enable this new interrupt level indicator value into the current status latch (refer to Figure 3-15). If another interrupt request is received, its priority level will be compared with the updated value now in the current status latch.

After servicing an interrupt request, it is the responsibility of the interrupt service routine to restore the nested priority table in the RAM and the current status latch in the 3214 to their former values. This can be accomplished by executing an I/O write (output) instruction to port FD_{16}.

It should be noted here that the interrupt mask we referred to earlier can be examined and/or updated by executing an I/O instruction to address FC_{16}. The detailed explanation of these three special internal control cycles, however, is postponed until the next section where we will deal with all such special program-controlled operations.

Interrupt timing is shown in Figure 3-16. Notice that the rising edge of the interrupt request input to the 8080 (INT) is referred to the ϕ_2 clock pulse and that INT is not reset until DBIN and INTA/ are encountered, thus ensuring that, if interrupts are enabled, the 8080 will recognize the interrupt request after the current instruction is executed.

NOTE: The current interrupt logic is implemented with a combination of standard TTL logic, and the 3214 interrupt control unit, but will be completely replaced by a new INTEL LSI device, the 8259, at a later date. This will result in a major reduction in the complexity of the module, but some incompatiblities may exist. The software can be made independent of these incompatibilities, if appropriate steps are taken now in anticipation of future changes.

During interrupt operations, the following differences exist:

1. When using the 8259 device, a 3-byte CALL instruction will replace the 1-byte restart (RST) instruction, but the same interrupt vector addresses will be used.
2. When using the 8259, a lockout of further interrupts will occur at the onset of an interrupt acknowledge from the 8080 (INTA). The current logic performs the lockout on acceptance of the first interrupt. A difference of up to 4-μs may develop, but is considered to be inconsequential to any real-time application envisioned for this module.

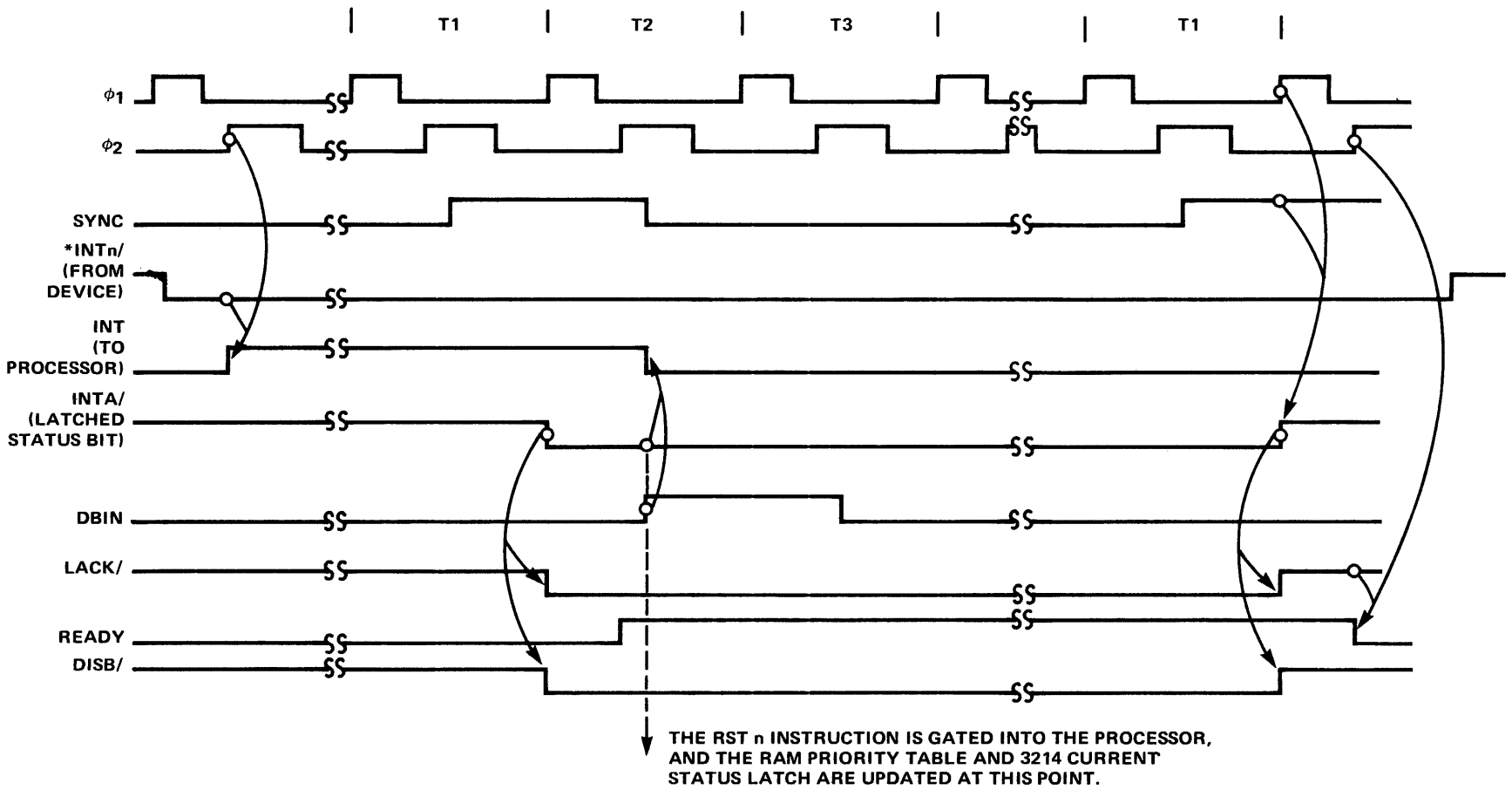
During initialization, a programmed startup procedure is required. Initialization of the current logic requires a system reset (external input) and an output to the mask register (port 'FC_{16}'). The 8259 device, to be used in the future, will require a programmed reset in addition to the system reset. The sequence required consists of outputting a data byte equal to '12_{16}' to port 'FD_{16}', followed by an output of '00' to port 'FC'. If this operation precedes all other interrupt operations, the operation of the current logic will not be affected. The following sequence may then be used to initialize both current and future implementations of the interrupt logic:

```
      .
      .
      .
MVI   A, 12
OUT      FD
MVI   A, 00
OUT      FC
      .
      .
      .
```

When using an 8259 device, interrupt servicing must include restoration of the previous operating level before a return to the interrupted program. The restoration is achieved by outputting a specific byte to I/O port 'FD_{16}' (the specific byte is an 8259 requirement). During execution of the output instruction, the interrupts must be disabled to avoid further interruption at the serviced level before the return can be completed. This could result in stack overflow. An example of an acceptable termination sequence is given below:

```
      .
      .
      .
DI
MVI   A, 20
OUT      FD
POP      PSW
EI
RET
      .
      .
      .
```

Because of the full nesting capabilities of both current and future designs, interrupts may be enabled



*WHERE n IS THE HIGHEST PRIORITY INTERRUPT LEVEL CURRENTLY REQUIRING SERVICE.

Figure 3-16. Interrupt Timing

as a service routine is entered. An example of a service routine that saves and restores all of the CPU registers is given below-

```
EI
PUSH    PSW   }
PUSH    B     }  Save all registers
PUSH    D     }
PUSH    H     }
  .
  .
  .
POP     H     }
POP     D     }  Restore registers
POP     B     }
DI                Disable interrupts
MVI     A, 20 }  Restore previous operating
OUT     FD    }  level and restore A and
POP     PSW      Flog register
EI               Enable interrupts
RET              Return
```

3.4.7 SPECIAL INTERNAL CONTROL CYCLE

The Central Processor Module has provisions for performing special internal control operations by executing I/O instructions to dedicated addresses (FC_{16} to FE_{16}). The internal operations are:

(1) *Define and store interrupt mask.* To define and store the interrupt mask, an 8-bit dataword should be output to port FC_{16} (OUT ØFCH). Each of the 8 bits correspond to one of the eight interrupt levels. If the bit for a particular level is a "1", that level is disabled. The 8-bit interrupt mask is stored in an 8212 latch within the interrupt logic (see sheet 3 of the module schematic, Figure 3-19). Also refer to the NOTE at the end of Section 3.4.6.

(2) *Read the interrupt mask.* The 8-bit interrupt mask stored in the 8212 latch is gated through another 8212 section and input to the 8080 processor when an input instruction to address FC_{16} (IN ØFCH) is executed.

(3) *Restore interrupt priority level.* Recall that the 3-bit level indicator for the interrupt currently being serviced is pushed onto the nested priority table in the 3101 RAM element and stored in the 3214 interrupt control unit's current status latch. After servicing an interrupt, the program must restore the nested priority table and the current status latch by executing an output instruction to address FD_{16} (OUT ØFDH). Execution of the OUT ØFDH instruction decrements the 74191 counter that addresses the 3101 RAM and enables the new 3-bit level indicator that is output by the RAM into the 3214's current status latch. Thus, the interrupt logic is restored; that is, it is now capable of responding to the next interrupt request. Also refer to the NOTE at the end of Section 3.4.6.

(4) *Override loss of the bus.* If it is necessary to quarantee that the CPU module not lose control of the system bus, the override function can be invoked by executing an output instruction to address FE_{16} (OUT ØFEH). If data bit 0 is a "1" when OUT ØFEH is executed, the override flip-flop is set; OVERRIDE/ goes true (refer to sheet 2 of the module schematic, Figure 3-15). OVERRIDE/ is gated through to the J input of the busy flip-flop. While OVERRIDE/ is true, the bus control logic is prevented from relinquishing control of the bus. If data bit 0 is a "0" when OUT ØFEH is executed, or if the initialization (INIT/) signal occurs, the override flip-flop resets and OVERRIDE/ goes false. The CPU Module must reset the override capability when it is finished with exclusive use of the bus, by executing an OUT ØFEH instruction with data bit 0 equal to "0".

The special control operations proceed exactly like a normal I/O cycle, with the following exceptions (refer to sheet 1 of the module schematic, Figure 3-19):

- The address lines are decoded by logic on the CPU module. If one of the dedicated addresses is detected, a 3205 decoder issues one of the following signals: FD, FC or FE/, indicating the address. These signals are used, in conjunction with IOWT/ or IMASK/, to effect the desired control operation. IOWT/ merely specifies that an output instruction is being executed; it is referred to the write strobe WR/. IMASK/ indicates that an input instruction is being executed; it is referred to the input strobe DBIN.

- If a dedicated address is detected and IOWT/ or IORD/ occur (indicating a special internal control operation), a local acknowledgement (LACK/) signal is generated. LACK/ performs the same function as either of the external acknowledgements, XACK/ or 8080/. Because no external device responds to any of the dedicated addresses, LACK/ must be generated to enable READY and allow the processor to proceed to state T3.
- Because no data is actually being transferred to/from the CPU Module during special control cycles, it is important that the two 8226 bidirectional bus drivers that gate data on/off the external data bus be disabled during these special operations. If an I/O instruction to hexadecimal address FC, FD or FE is executed, or if the interrupt status bit (INTA/) is true, the disable bus (DISB/) signal is generated. DISB/, as its name implies, disables the two 8226 bus drivers by driving the chip select ($\overline{CS}$) inputs inactive (high).

Timing for the internal control cycles is shown in Figures 3-17 and 3-18.

3.4.8 CENTRAL PROCESSOR MODULE SCHEMATIC

Figure 3-19 provides a complete schematic drawing (4 sheets) of all logic on the Central Processor Module.

3.5 UTILIZATION: CENTRAL PROCESSOR MODULE

This section provides information on utilization of the Central Processor Module outside of the INTELLEC MDS System.

3.5.1 INSTALLATION

In installing the Central Processor Module, the user must take account of:

(a) environmental extremes
(b) mounting considerations
(c) electrical connections
(d) power requirements
(e) signal requirements
(f) jumper connections

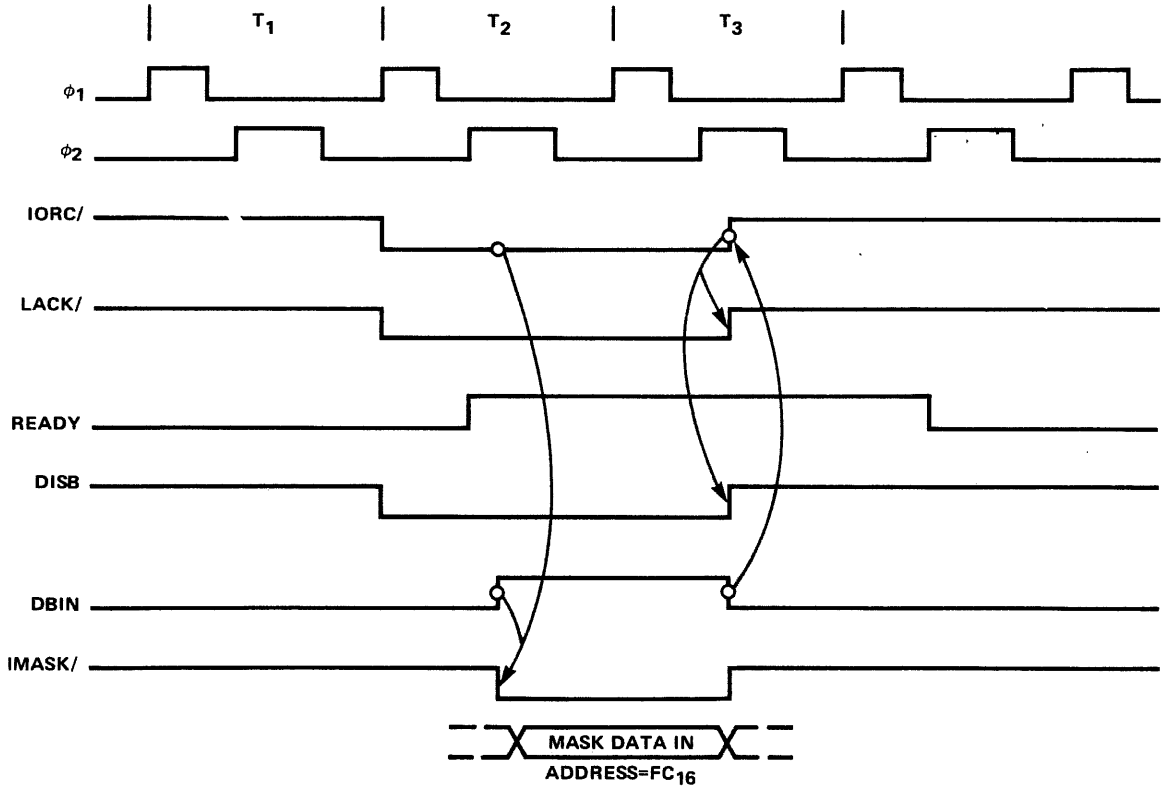


Figure 3-17. Read Interrupt Mask Control Cycle Timing

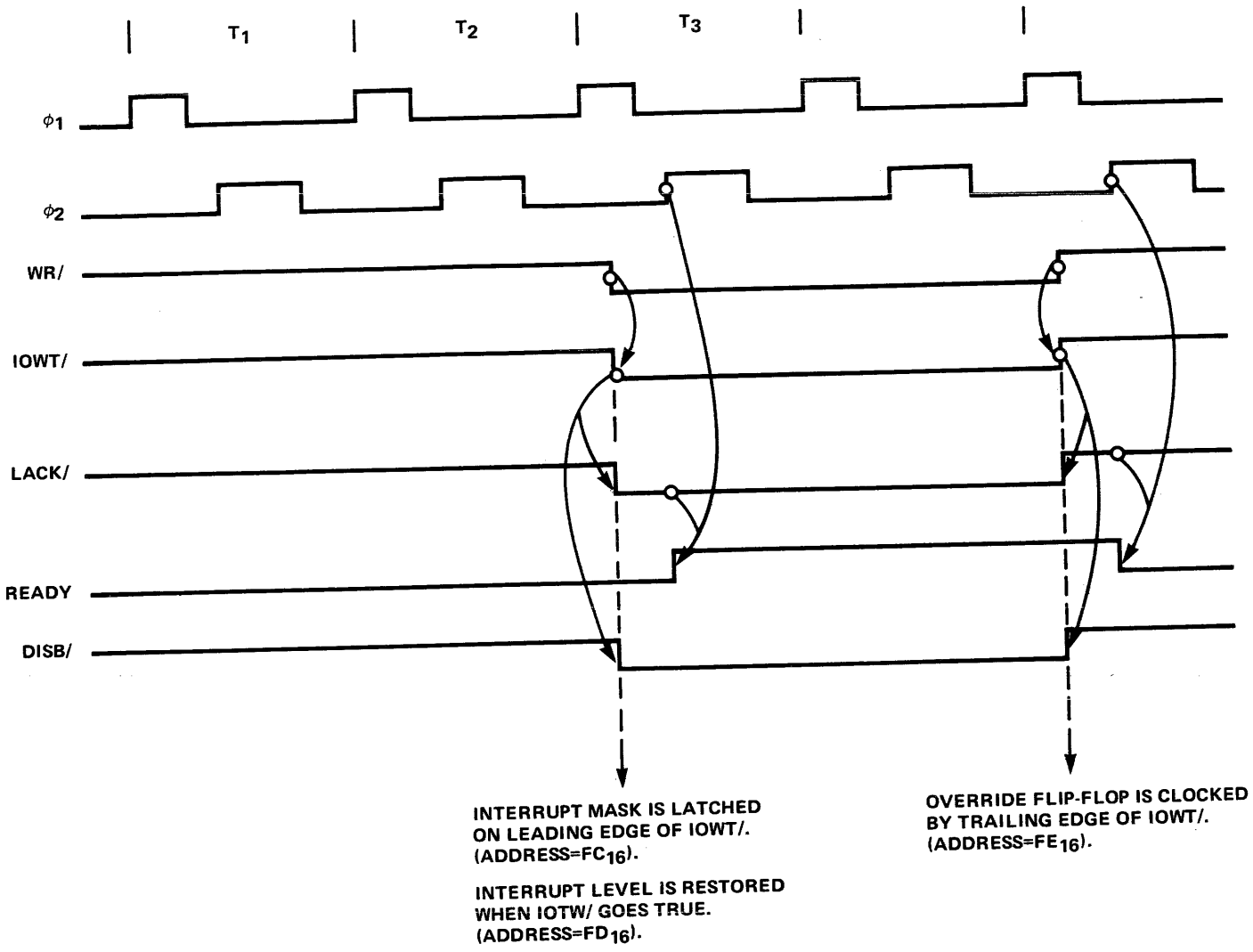


Figure 3-18. Internal Control (Output) Cycle Timing

Environment

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must, therefore, be maintained within the limits of 0° to 70° Centrigrade. Exercise caution in locating the module, giving particular attention to radiant and conducive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

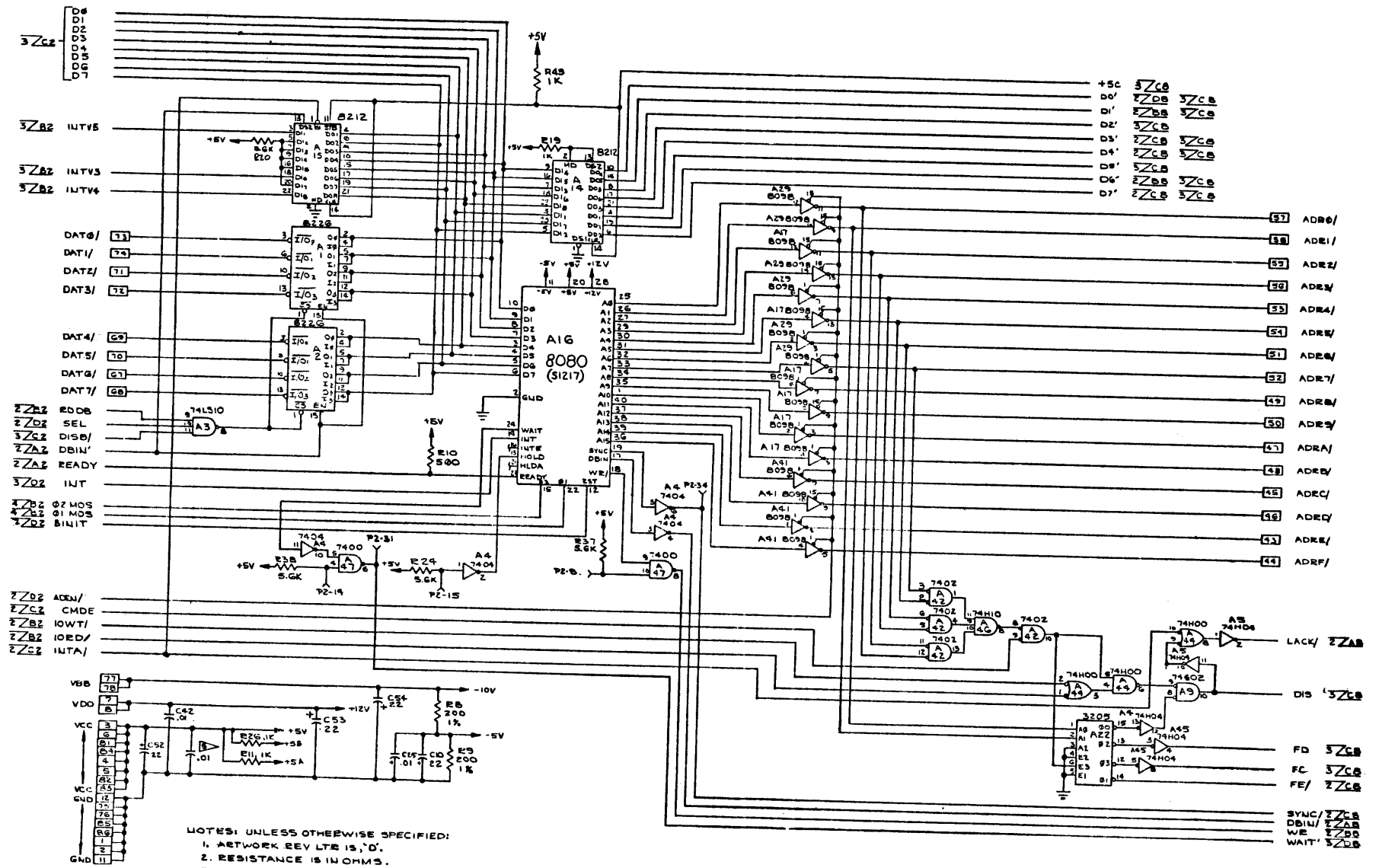
Relative humidity should not exceed 90%, noncondensing.

Mounting

Avoid locating the module near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections on the board, resulting in abnormally high noise levels or outright failure of the assembly.

Dimensions of the module are 12-in. X 6.75-in. Be sure to allow enough additional clearance to ensure adequate cooling.

The module is designed to plug directly into two standard double-sided PC edge connectors; one an 86-pin connector, the other a 60-pin auxiliary connector. The connectors will serve as a mounting, as



NOTES: UNLESS OTHERWISE SPECIFIED:

1. ARTWORK REV LTR IS 'D'.
2. RESISTANCE IS IN OHMS.
3. CAPACITANCE IS IN MICRO FARADS.
4. JUMPER J1 A & B IS IN FOR SERIAL PRIORITY; AND OUT FOR PARALLEL PRIORITY.
5. THE FOLLOWING ARE BETWEEN GND & +5; C11-18,20,23,24,26-30, 34-41 & 43-51.
6. EDGE CONNECTOR PINS ARE, P1.

TITLE	SCHEMATIC CPU MODULE		
SIZE	DEPT	DRAWING NO.	REV
D	410	2000342	D

Figure 3-19. Central Processor Module Schematic

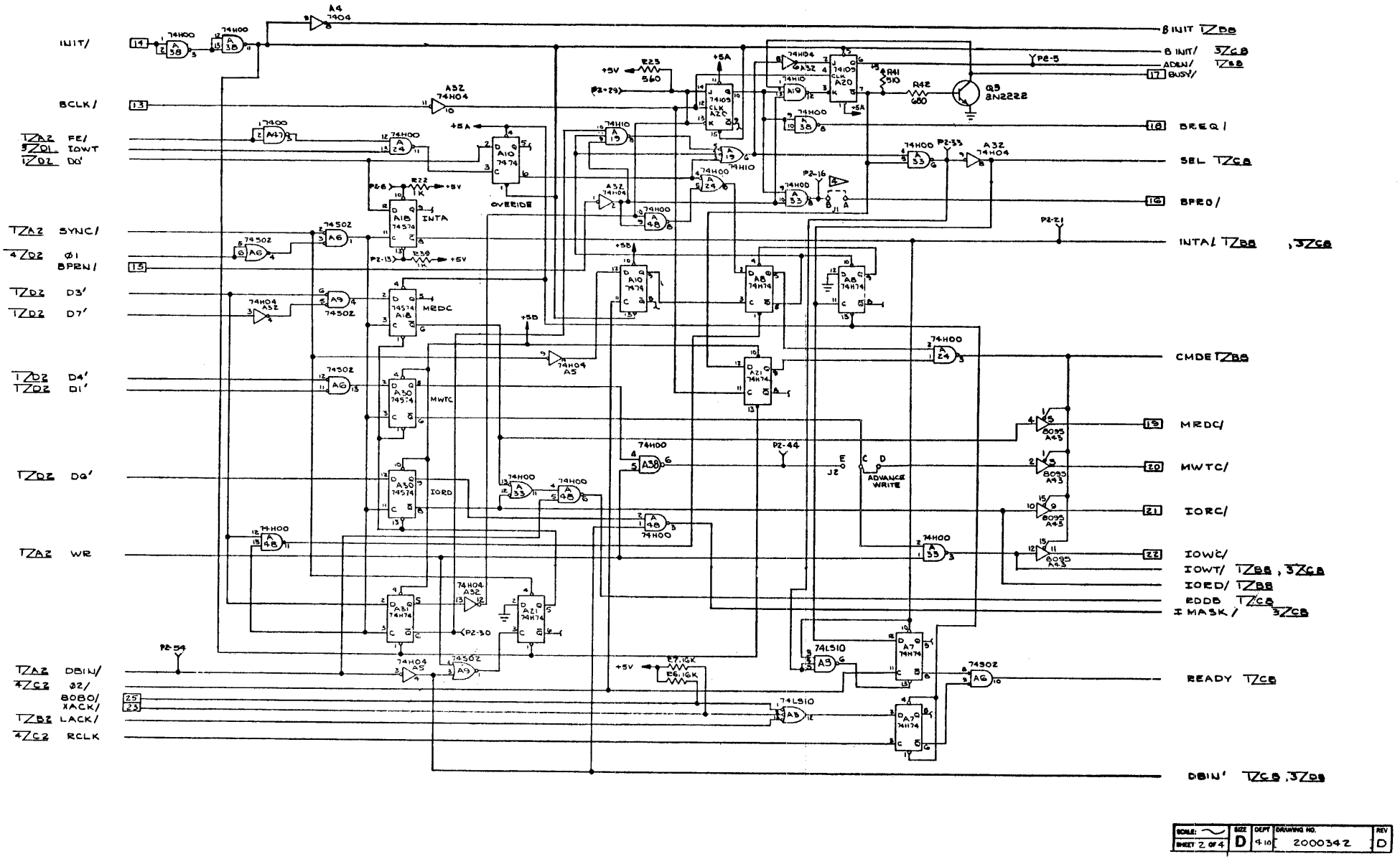


Figure 3-19. Central Processor Module Schematic (Sheet 2 of 4)

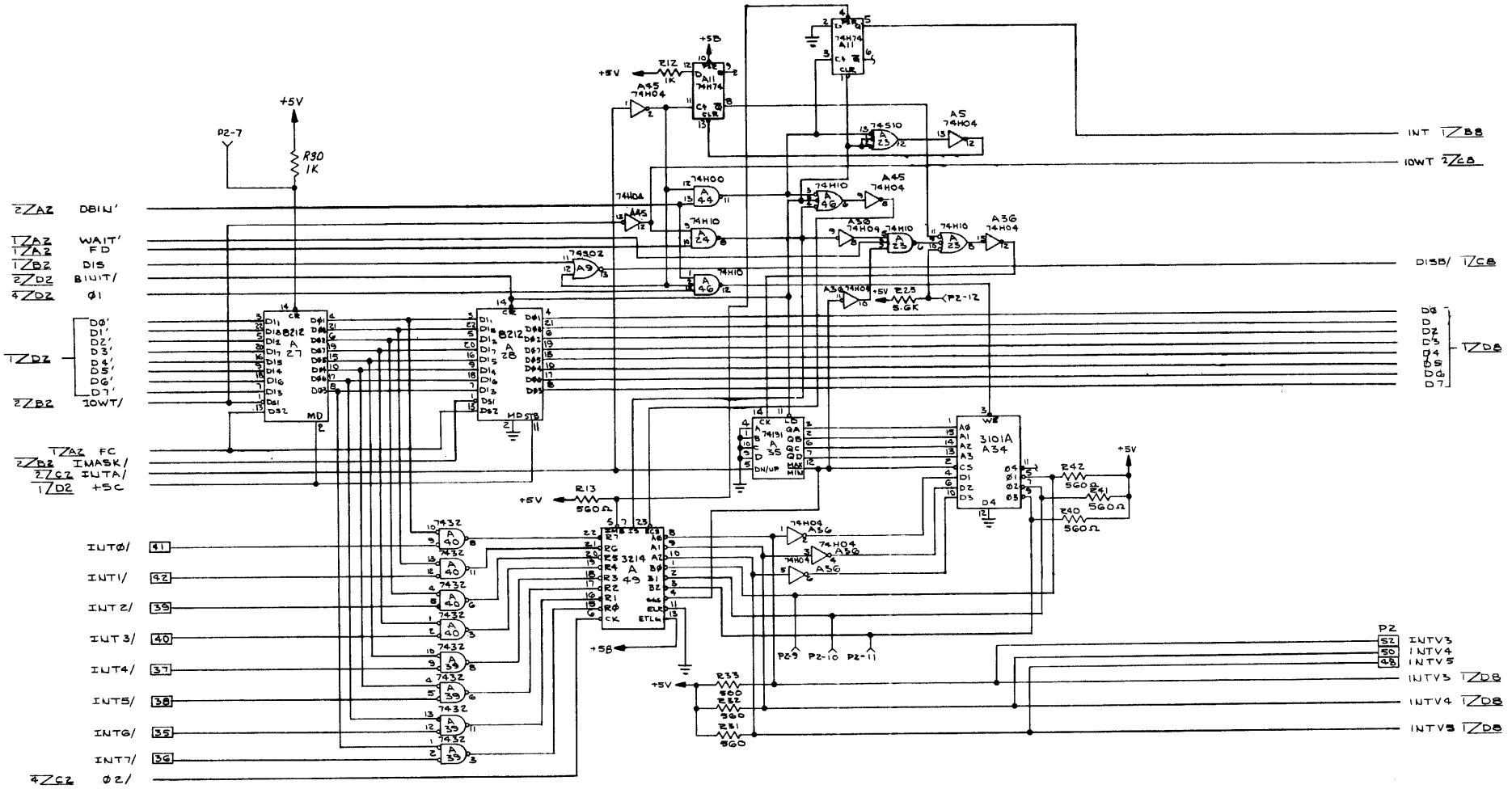


Figure 3-19. Central Processor Module Schematic (Sheet 3 of 4)

SCALE:	SIZE	DEPT	DRAWING NO.	REV
SHEET 3 OF 4	D	410	2000342	D

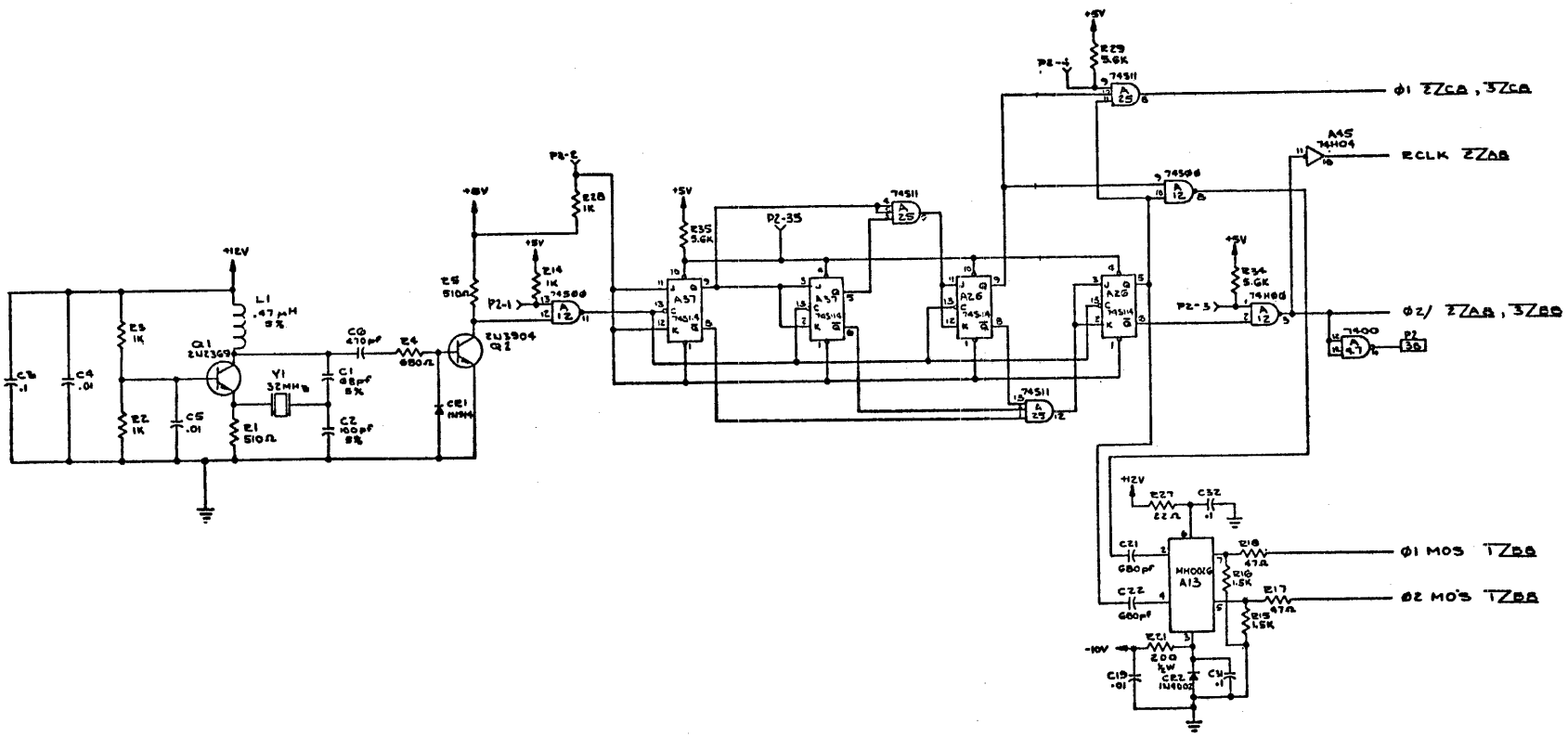


Figure 3-19. Central Processor Module Schematic (Sheet 4 of 4)

well as an electrical junction, if the envrionment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the assembly be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the board, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the components on the module.

Electrical Connections

The basic power and control connections to the CPU Module are made through a standard 86-pin, double-sided PC edge connector (P1), 0.156-in. contact centers. Control Data Corp. VPB01E43-A00A1 is one suitable type. Pin allocations on this connector are given in Table 3-6 of Section 3.5.2. Additional signal connections are possible through an auxiliary 60-pin, double-sided PC edge connector (P2), 0.100-in. contact centers, as shown in Figure 3-20. Pin allocations on the auxiliary connector are given in Table 3-7 of Section 3.5.2.

Signal Requirements

All data and control functions appearing at the module edge connectors are at TTL levels. Electrical characteristics of the specific signal inputs and outputs and power inputs are given in Section 3.6.

Signal descriptions and connector pin allocations are given in Section 3.5.2.

Jumper Connections

A jumper pad (1-2) on the Central Processor Module (see sheet 2 of the module schematic) enables the bus priority out (BPRO/) signal (pin 16). When jumper 1-2 is connected, the bus priority in (BPRN/) signal will be serially passed via BPRO/ to the module in the rack slot adjacent to the Central Processor Module, unless, of course, the Central Processor Module requires control of the bus and captures BPRN (i.e., unless gate A33-8-9-10 is not activated).

BPRO/ allows implementation of a serial, bus priority scheme instead of the 8-level parallel scheme used in INTELLEC MDS Systems. A module's priority is a result of its relative position in the card rack; that module which is nearest to the source of BPRN/ has highest priority and that module which is farthest from the source has lowest priority. Note that a serial priority scheme is dependent on the physical presence of a board in each card position between the highest and lowest priority modules. Also keep in mind that resolution of bus exchanges may require more time than with a parallel scheme.

BPRO/ may also be used in conjunction with the 8-level parallel priority scheme to increase the maximum number of master modules to 16. Each odd card position, except #1, may be paired with the even position directly to the right. In this case, both excite the same bus request. Simultaneous requests are resolved by the serial priority circuits on each board. This implementation allows up to 16 masters but introduces a dependency upon the presence of the odd partner of a pair and the installation of a BPRO/ jumper. If a pair is used, the BPRO/ jumper must be *IN* on the odd slot but *OUT* on the even slot.

Another jumper pad, labeled "advanced write" (C-D-E), can be used to disable the advanced acknowledge feature described in Section 3.4.4. To disable the special advance acknowledge feature, disconnect jumper connection D-C and connect E-D. The 8080/ line must also be disabled on the CPU or memory modules.

3.5.2 PIN LIST: CENTRAL PROCESSOR MODULE

The following section describes connector pin allocations on the Central Processor Module. The pins and their designated signal functions for the bus connector (P1) are listed in Table 3-6. The same information for the auxiliary connector (P2) is listed in Table 3-7.

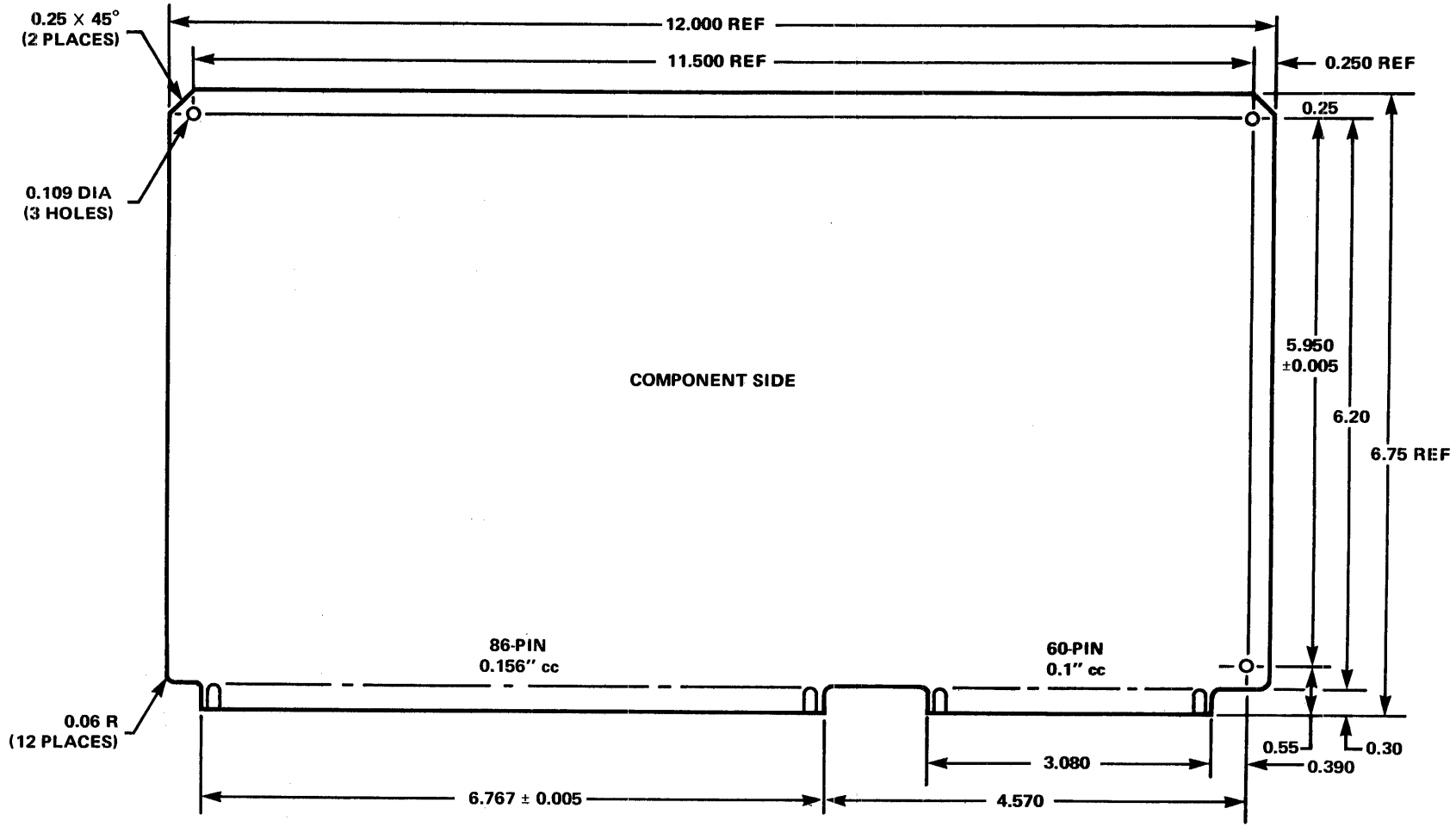


Figure 3-20. CPU Module Edge Connectors

Table 3-6

BUS CONNECTOR (P1) PIN LIST

PIN	SIGNAL	FUNCTION
1	GND	Ground
2	GND	
3	V_{CC}	Source power +5 VDC
4	V_{CC}	
5	V_{CC}	
6	V_{CC}	
7	V_{DD}	Source power +12 VDC
8	V_{DD}	
9		
10		
11	GND	Ground
12	GND	
13	BCLK/	Bus Clock
14	INIT/	Initialize system
15	BPRN/	Bus priority in
16	BPRO/	Bus priority out
17	BUSY/	Bus busy
18	BREQ/	Bus request
19	MRDC/	Memory read command
20	MWTC/	Memory write command
21	IORC/	I/O read command
22	IOWC/	I/O write command
23	XACK/	External acknowledge
24		
25	AACK/	Advanced 8080 acknowledge
26		
27		
28		
29		
30		
31		
32		
33		
34		
35	INT6/	External Interrupt Level Requests
36	INT7/	
37	INT4/	
38	INT5/	
39	INT2/	
40	INT3/	
41	INTØ/	
42	INT1/	
43	ADRE/	ADDRESS BUS

PIN	SIGNAL	FUNCTION
44	ADRF/	ADDRESS BUS
45	ADRC/	
46	ADRD/	
47	ADRA/	
48	ADRB/	
49	ADR8/	
50	ADR9/	
51	ADR6/	
52	ADR7/	
53	ADR4/	
54	ADR5/	
55	ADR2/	
56	ADR3/	
57	ADRØ/	
58	ADR1/	
59		
60		
61		
62		
63		
64		
65		
66		
67	DAT6/	DATA BUS
68	DAT7/	
69	DAT4/	
70	DAT5/	
71	DAT2/	
72	DAT3/	
73	DATØ/	
74	DAT1/	
75	GND	Ground
76	GND	
77	V_{BB}	Source power -10 VDC
78	V_{BB}	
79		
80		
81	V_{CC}	Source power +5V
82	V_{CC}	
83	V_{CC}	
84	V_{CC}	
85	GND	Ground
86	GND	

Table 3-7

AUXILIARY CONNECTOR (P2) PIN LIST

PIN	FUNCTION	PIN	FUNCTION
1	TP – CLOCK DISABLE	31	FP – RUN INDICATOR RUNI/
2	TP – COUNTER CLEAR	32	
3	TP – Ø2 DISABLE	33	TP – SELECTED
4	TP – Ø1 DISABLE	34	TP – SYNC
5	TP – ADDRESS ENABLE	35	TP – COUNTER SET
6	TP – INTERRUPT SET	36	
7	TP – MASK CLEAR	37	
8	TP – WR/ DISABLE	38	GP – BUFFERED CPU CLOCK
9	TP – 3101 01	39	
10	TP – 3101 02	40	
11	TP – 3101 03	41	
12	TP – PUSH/POP	42	
13	TP – INTA CLEAR	43	
14	TP – WAIT CONTROL	44	TP – RWCMD/
15	TP – HOLD CONTROL	45	
16	TP – BPRØ JUMPER	46	
17		47	
18		48	FP – INV5
19		49	
20		50	FP – INV3
21	FP – INTA/	51	
22		52	FP – INV4
23		53	
24		54	FP – DBIN/
25		55	
26		56	
27		57	
28		58	
29	GP – TRANSFER REQUEST	59	
30	FP – HALT INDICATOR HLTL/	60	

TP = TEST POINT ONLY
FP = FRONT PANEL INTERFACE FUNCTION
GP = GENERAL PURPOSE INTERFACE FUNCTION

3.6 OPERATING CHARACTERISTICS: CPU MODULE

This section provides detailed information concerning the AC and DC characteristics of the CPU Module.

3.6.1 AC CHARACTERISTICS

Detailed timing diagrams for memory, I/O and Bus exchange operations are provided in Figures 3-21 through 3-25. Table 3-8 provides design limits for CPU module outputs and requirements for its inputs. These values are theoretical limits based on a "worst-on-worst" case analysis using vendor information and approximations where necessary. Approximations include establishing non-zero propagation delay minimums and extended delays if capacitive loading exceeds vendor ratings. In all such cases, approximations are conservative (e.g., 2 ns minimum for standard TTL, 4 ns minimum for tri-state turn-offs or turn-ons). Rise and fall times are assumed to be zero unless a three-state high impedance state or open collector circuit is involved.

The timing is compatible with the INTELLEC MDS Bus specifications, with two exceptions. These exceptions are included to allow the 8080 processor to run at full speed for memory read and write operations. Neither of the exceptions severely limits the flexibility of the module. In fact, they may be optionally overriden to achieve full compatibility with the bus if speed is not critical.

The exceptions cited above are the use of an advanced acknowledge (AACK/), and the delay to stable data during a memory write operation (t_{DD}). Both stem from the manner in which READY input is sampled by the 8080 CPU. The MDS Bus specification requires that an acknowledge be returned only if the "slave" device has both received a command and completed the necessary data transfer. The 8080, however, samples READY prior to generating the leading edge of its two basic commands, DBIN for read and WR/ for write. If these commands are used, the first 8080 sampling will find an MDS slave device "not ready", since the command has yet to reach the bus. An example of this is shown in Figure 3-23, I/O write timing. The I/O write timing uses WR/ to qualify the bus command and the first sampling point (▷ of figure 3-23) occurs before the leading edge of the command. The consequence is an unavoidable 500-ns delay, regardless of the "slave" device's response time. This may result in up to 33% and 14% decreases in the operating speeds of memory read and write operations, respectively.

To avoid the degradation cited above, commands are initiated earlier in the 8080 cycle and the advanced acknowledge, unqualified by stable data, is allowed. This permits a "slave" device to return a ready condition earlier than the 8080 sampling point. Any use of the 8080/ advanced acknowledge or advanced memory write must consider the absolute timing requirements as specified in this section.

3.6.2 DC CHARACTERISTICS

The DC characteristics for all INTELLEC MDS Bus functions provided by this board are given in Table 3-9. They are derived from vendor specifications and calculated values if passive loading exists. Capacitance values are approximations only.

Power requirements are cited below:

		TYP	MAX
V_{CC}	+5VDC ± 5%	2.4A	3.0A
V_{DD}	+12VDC ± 5%	0.07A	0.1A
V_{BB}	−10VDC ± 5%	0.07A	0.1A

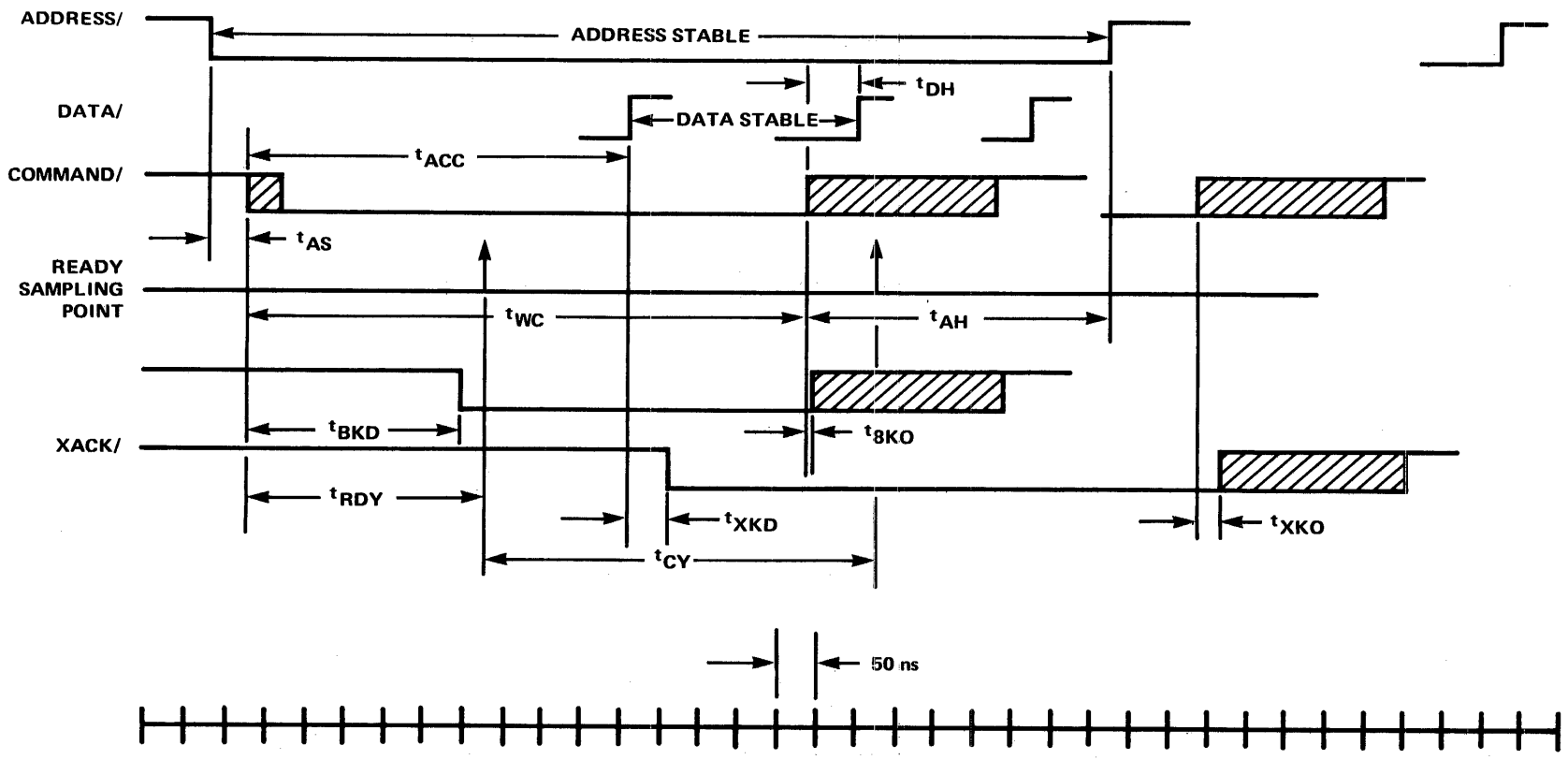


Figure 3-21. Memory and I/O Read Timing (Continuous Bus Control)

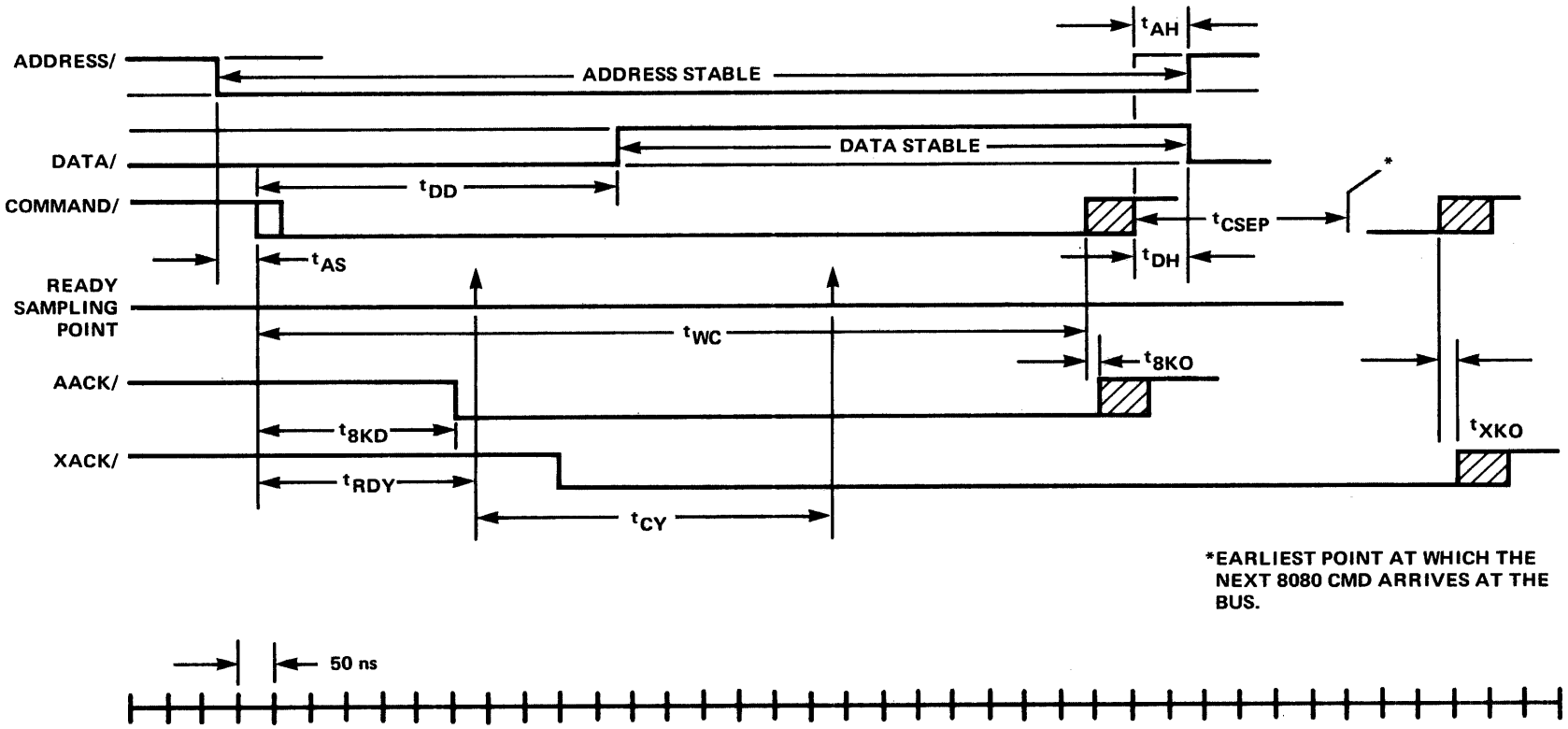


Figure 3-22. Memory Write Timing (Continuous Bus Control)

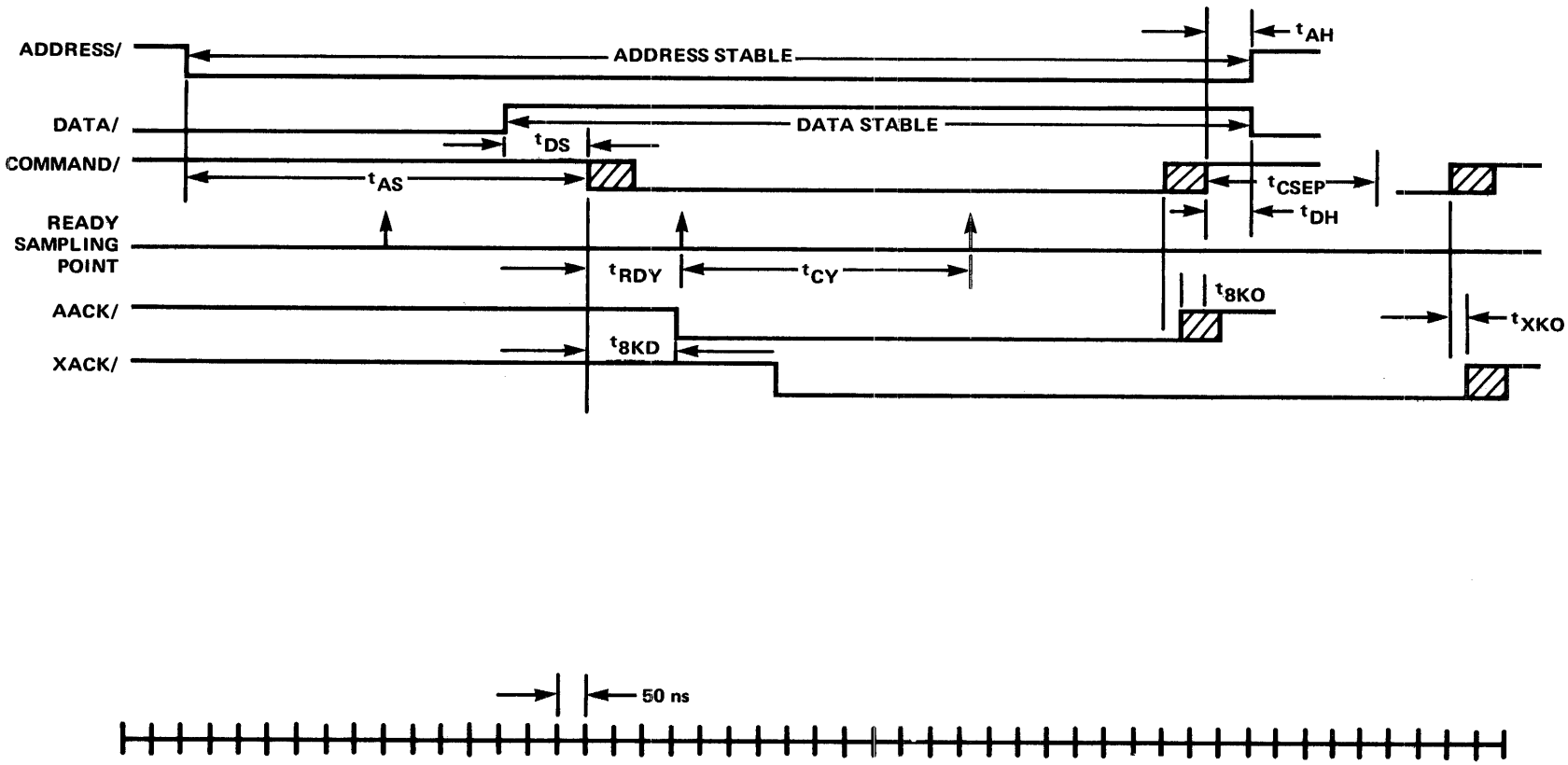


Figure 3-23. I/O Write Timing (Continuous Bus Control)

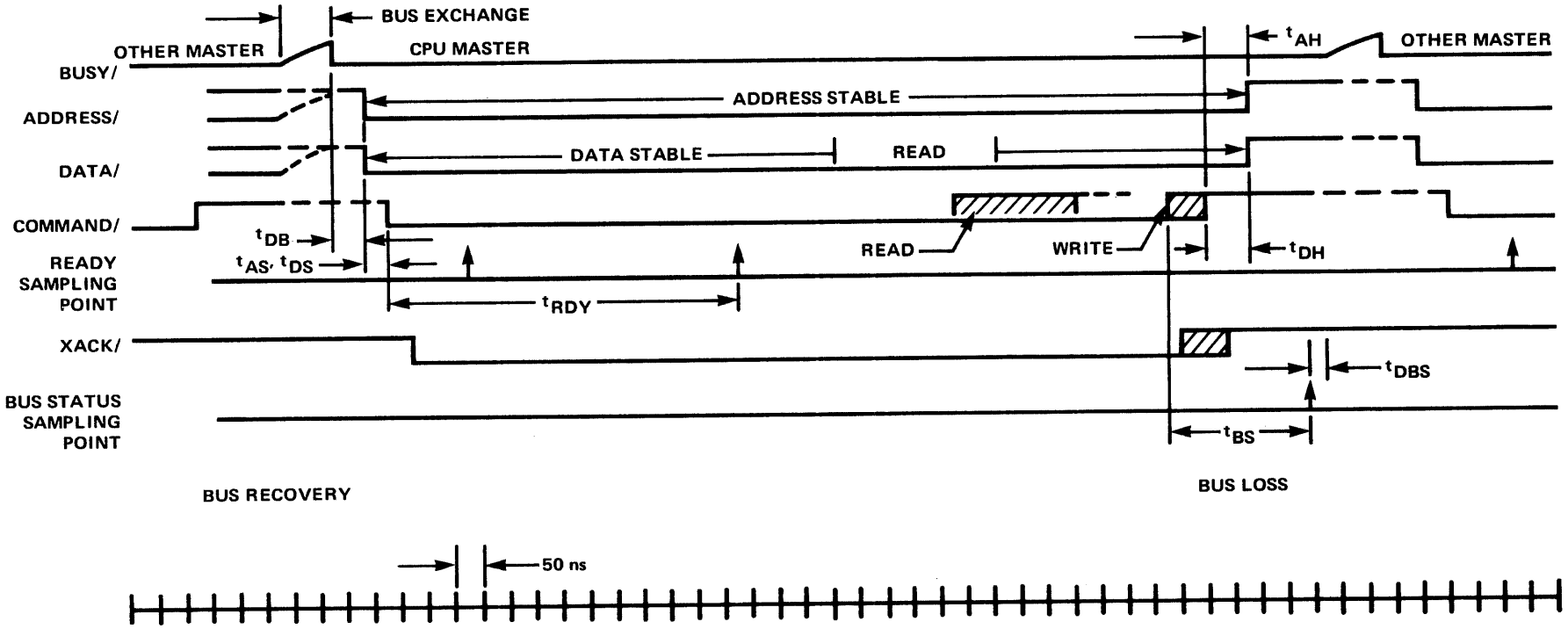


Figure 3-24. Bus Exchange (Read/Write)

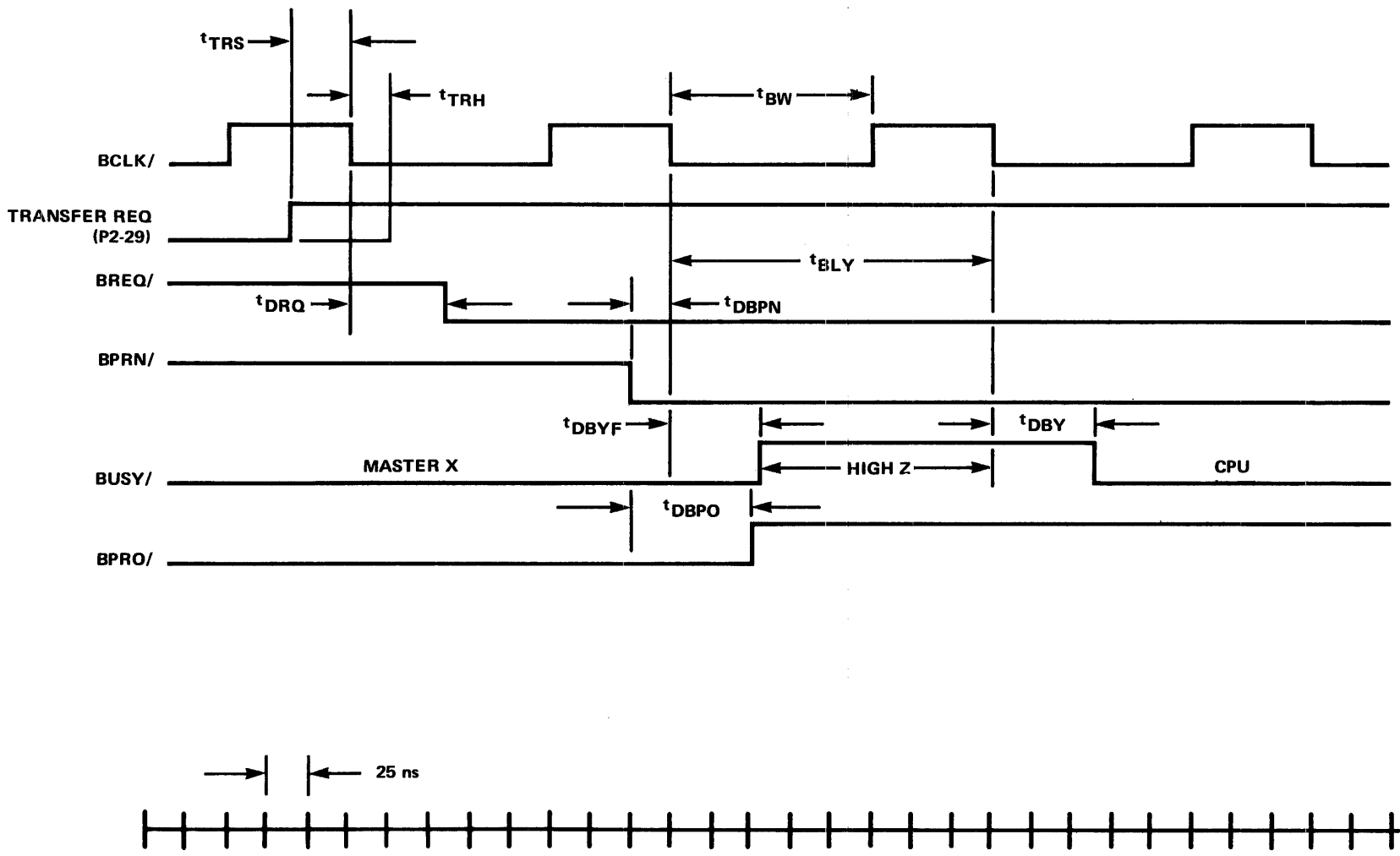


Figure 3-25. Bus Exchange

Table 3-8

CPU MODULE AC CHARACTERISTICS

PARAMETER	OVERALL		CONTINUOUS BUS CONTROL						BUS EXCHANGE		DESCRIPTION	REMARKS
			READ		MEMORY WRITE		I/O WRITE		MEMORY WRITE			
	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)		
t_{AS}	50		50		50		600		50		Address Setup Time to Command	
t_{AH}	70		150		70		70		70		Address Hold Time	▷1 Read Cycle is Max Case
t_{DS}	50				See t_{DD}		180		50		Data Setup Time to Command	▷2 Max Assumes No Acknowledge Delays
t_{DH}	70		150		70		70		70		Data Hold Time	
t_{DD}		500				500					Data Delay During Memory Write	
t_{RDY}	0<	1010	140		140		0<	90	495	1010	Ready Sampling Point	
t_{CY}	495	505									Cycle Time	Output Limits
t_{SEP}	180	▷2	270	▷2	180	▷2	180	▷2			Command Separation	
t_{WC}	490	1230	650	950	1100	1230	490	510			Command Width	
t_{DBS}									0	105	Bus Sample to Exchange Initiation	
t_{BS}	200	720							200	▷1 720	Bus Sampling Point Delay	
t_{DB}		55								55	Data and Address Turn On Delay	
t_{DRQ}		35								35	Bus Request Delay	
t_{DBY}		65								65	Bus Busy Turn On Delay	
t_{DBYF}		40								40	Bus Busy Turn Off Delay	
t_{DBPO}		20								20	BPRO/ Serial Delay from BPRN/	
t_{ACC}		450		450							Read Access Time	
t_{8KD}		140		140		140					8080 ACK Response Time for No Delay	Input Requirements
t_{8K0}		70		70		70		70			8080 ACK Turn Off Delay	
t_{XKD}	50		50			50		50			XACK Delay from Valid Data or Write	
t_{XK0}		70		70		70		70			XACK Turn Off Delay	
t_{BCY}	100								100		Bus Clock Cycle Time	
t_{BW}	25								25		Bus Clock Low and High Periods	
t_{TRS}	10								10		Transfer Request Setup Time	
t_{TRH}	10								10		Transfer Request Hold Time	
t_{DBPN}	30								30		Priority Input Setup Time	

CPU MODULE DC CHARACTERISTICS

SIGNALS	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNITS
ADRØ/–ADRF/ ADDRESS	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = 5.2 mA	2.4		V
	V_{IL}	Input Low Voltage	T_A = 25°C		0.8	V
	V_{IH}	Input High Voltage		2.0		V
	V_{IL}	Input Current at Low V	V_{IN} = 0.4		–1.64	mA
	V_{IH}	Input Current at High V	V_{IN} = 2.4V		80	μA
	C_L	Capacitive Load			15	pF
MROC/, MWTC/ IORC/, IOWC/	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = –5.2	2.4		V
	I_{LH}	Output Leakage High	High Z V_F = 0.4		–40	μA
	I_{LL}	Output Leakage Low	High Z V_R = 2.4		40	μA
	C_L	Capacitive Load			15	pF
DATØ/–DAT7/	V_{OL}	Output Low Voltage	I_{OL} = 25 mA/50 mA		0.45/0.6	V
	V_{OH}	Output High Voltage	I_{OH} = –2.4 mA	2.4		V
	V_{IL}	Input Low Voltage			1	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		–0.25	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25		–1	mA
	I_{LH}	Output Leakage High	High Z V_F = 0.45V		–0.1	mA
	I_{LL}	Output Leakage Low	High Z V_R = 5.25		0.1	mA
	C_L	Capacitive Load			15	pF
INTØ/–INT7/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		–1.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		40	μA
	C_L	Capacitive Load			5	pF
INIT/, BCLK/ BPRN/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		–2.0	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		50	μA
	C_L	Capacitive Load			5	pF
XACK/, AACK/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		–2.3	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		50	μA
	C_L	Capacitive Load			6	pF
BUSY/	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = 5.2 mA	2.4		V
	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		–2.0	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		80	μA
	C_L	Capacitive Load			15	pF

Table 3-9

CPU MODULE DC CHARACTERISTICS (continued)

SIGNALS	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNITS
BREQ/, BREO/	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = –0.5 mA	2.4		V
	C_L	Capacitive Load			5	pF

Chapter 4
FRONT PANEL CONTROL MODULE

The standard INTELLEC MDS System includes a simple, but functional front panel that indicates the current status of the system (running or halted), and allows an operator to load a bootstrap program (firmware implemented), reset the entire system or assert an interrupt request on any one of eight interrupt levels. The front panel is shown in Figure 4-1.

The Front Panel Control Module, as its name implies, controls the front panel in the INTELLEC MDS System.The module drives the INTERRUPT, RUN and HALT indicators, and responds to the INTERRUPT, BOOT and RESET switches. The 256-byte bootstrap program is actually stored in a PROM on the board. The module's capabilities are not, however, restricted to controlling the front panel. The module provides the system with the following additional features:

- Eight-level parallel bus priority network that resolves all requests for control of the bus, on the basis of relative priority.

- Real-time clock that sets a status bit and generates an interrupt request at 1-ms intervals; the interrupt request, however, can be disabled under program control.

- Failsafe scheme that can be used to prevent the system from stopping because a nonexistent memory location or I/O port was addressed. After waiting 10 ms, the failsafe logic generates the necessary acknowledge signal, asserts an interrupt request and flashes an indicator on the module. This feature can be very useful during program development and

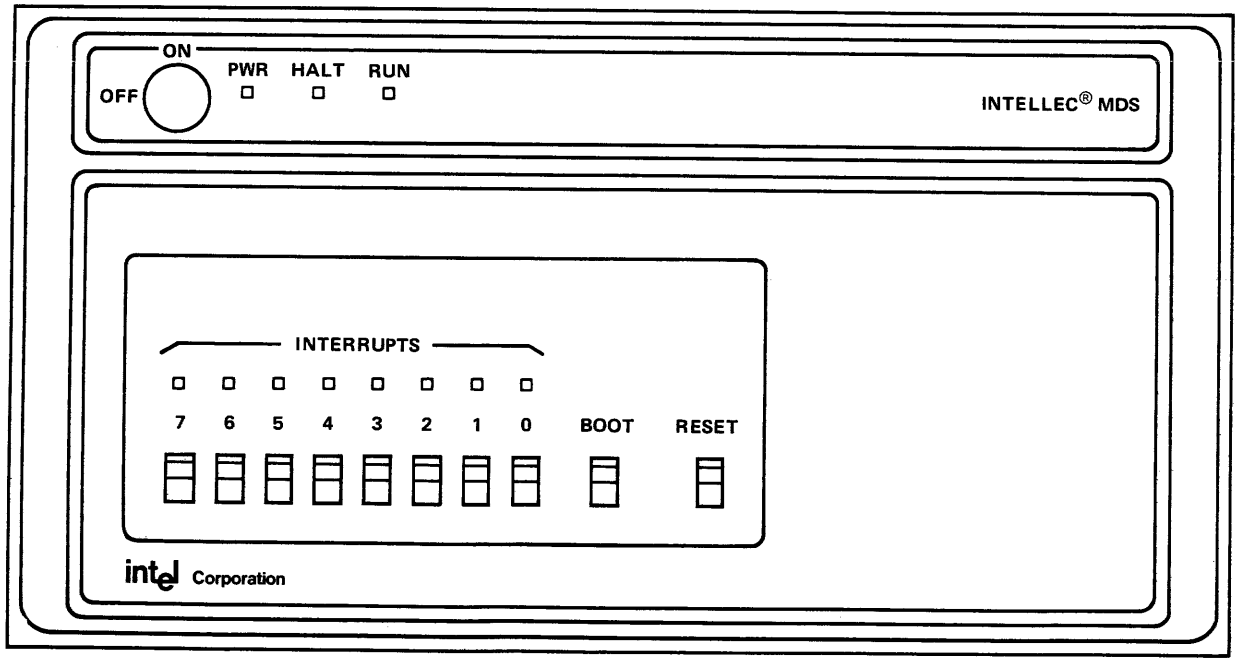


Figure 4-1. INTELLEC® MDS Front Panel

debugging. The acknowledge and interrupt portions can be easily disabled by disconnecting two solderless jumper pads if the features are not required.

While the Front Panel Control Module has been tailored for use in the INTELLEC MDS System, it, like all INTELLEC modules, is available independently on an OEM basis.

4.1 FUNCTIONAL ORGANIZATION OF THE FRONT PANEL CONTROL MODULE

For descriptive purposes, the Front Panel Control Module can be divided into six functional units, as shown in Figure 4-2:

(1) Bus priority logic
(2) Bootstrap logic
(3) Interrupt switch/indicator logic
(4) Reset switch logic
(5) Real-time clock
(6) Failsafe logic

The *bus priority logic* resolves bus contention for up to eight master modules. The logic monitors eight bus request (BREQ) lines, arbitrates all requests in parallel, and outputs eight bus priority (BPRN) lines. Only one BPRN line will be active at any given time; that is, the BPRN line associated with the highest priority module which is requesting use of the bus. Only a module which has been granted control of the bus can initiate transfers via the bus. The bus priority logic also generates the 9.8 MHz bus clock (BCLK/) signal which provides a timing reference for the bus control sections of the various master modules.

The *bootstrap logic* contains a special purpose PROM which occupies memory locations ØØØØ to $ØØFF_{16}$, when enabled by a front panel control switch (BOOT). The bootstrap program stored in the PROM can be used to initialize the system with a minimum of operator control required. The bootstrap program initializes the registers in the CPU and transfers control to the System Monitor program. If desired, a customized bootstrap program can be implemented by replacing or reprogramming the PROM. When enabled, the PROM effectively replaces all other memory in the designated address space (ØØ-FF_{16}) for all read operations. Write operations will operate on read/write memory at these locations. The PROM responds to access commands like any standard memory element; when it recognizes an address and memory read (MRDC) command, it outputs the appropriate data byte along with an external acknowledge (XACK), indicating that the data byte is on the data lines.

The *interrupt switch/indicator logic* accepts interrupt requests which are manually initiated at the switches on the front panel (see Figure 4-1), stores the requests until acknowledged by the CPU, and outputs the request(s) to the Central Processor Module. This logic also drives the interrupt indicators on the front panel. When an interrupt switch is pressed, the logic lights the corresponding indicator. Note that the Front Panel Control Module does not arbitrate multiple requests according to priority; this is done at the Central Processor Module.

When the RESET switch on the front panel is depressed, the *reset logic* generates an initialization (INIT/) signal which resets all logic on the Front Panel Control Module. INIT/ is also made available to all other modules, via bus pin 14. The INIT/ pulse is approximately 1-ms wide.

A *real-time clock* is implemented on the Front Panel Control Module. At 0.9765-ms intervals a status bit is set and an interrupt request can be generated on interrupt level 1. The real-time clock interrupt can be disabled or enabled under program control. The status bit can be input to the primary master upon execution of an input to port FF_{16} instruction. The status bit is reset when it is interrogated.

Because all external references to memory or I/O devices proceed asynchronously with respect to the processor, a positive acknowledgement scheme is used to inform the processor when the external element is ready (i.e., the processor idles in a wait state until either the XACK/ or AACK/ signal is returned). With such a positive acknowledge scheme, however, it is possible to halt the system indefinitely if a non-existent I/O port or memory location is addressed. To avoid this, a user can enable (by connecting a jumper pad) the *failsafe* timer on the Front Panel Control Module. This timer is triggered at the beginning of each command. If the timer timesout (i.e., if no new

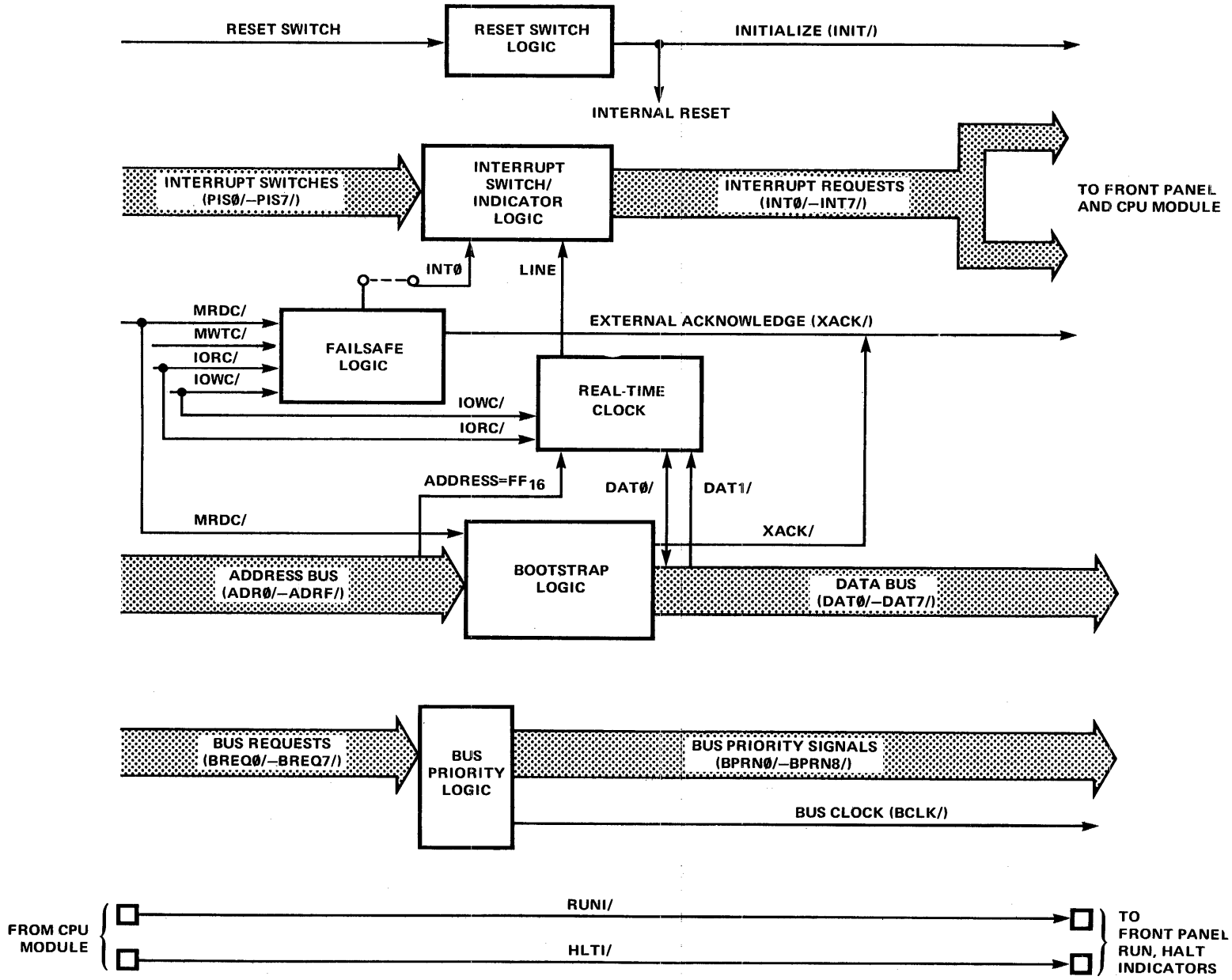


Figure 4-2. Front Panel Control Module Functional Block Diagram

command is received within approximately 10 ms), the failsafe logic asserts an acknowledge (XACK/) signal which allows the command cycle to be completed.

When the failsafe acknowledge is required to complete a read operation, the Front Panel Control Module presents all zeroes on the data bus. This transfer of erroneous data may, of course, result in system failure. It is anticipated, however, that in many instances the system will continue to run, thus facilitating the debugging of the related hardware or software problem.

If the proper solderless jumper pads in the failsafe logic are connected, a timeout will cause an interrupt request on level 2 and/or will flash the bus timeout display on the module, thus further facilitating system debugging.

You will notice in Figure 4-2 that RUNI/ and HLTI/ signals are shown passing through the Front Panel Control Module. These signals from the CPU Module, indicating that the CPU is running or halted, are used to drive the RUN and HALT indicators on the front panel (see Figure 4-1). The absence of both RUN and HALT indicates that the CPU is in a wait state.

4.2 FRONT PANEL CONTROL MODULE: THEORY OF OPERATION

Because the logic on the Front Panel Control Module performs several relatively unrelated functions, we will divide the detailed theory of operation description into six sub-sections, each dealing with one of the functional units defined in the previous section.

The Front Panel Control Module accepts/transmits signals, data and power through three different PC edge connectors:

- J1 Front panel connector (to/from the front panel)
- P1 Bus connector (to/from the system bus)
- P2 Auxiliary connector (to/from the auxiliary bus)

To avoid any ambiguity when referring to connector pins in subsequent paragraphs, we will always list the connector as well as the pin whenever such references are required; for example, P1-14 refers to pin 14 on connector P1. Pin lists for the three connectors are provided in Section 4.3.2.

The schematic (2 sheets) for the Front Panel Control Module is provided in Figure 4-8 (located in Section 4.2.7).

4.2.1 BUS PRIORITY LOGIC

The bus priority logic consists of a 74148 eight-to-three priority encoder, a 3205 three-to-eight decoder, a 19.67-MHz crystal-controlled clock oscillator, a 7474 D-type flip-flop, and various gating circuits, as shown on sheet 2 of the module schematic, Figure 4-8.

Bus request (BREQ/) lines from up to eight master modules are applied to the inputs of the 74148 priority encoder. The 74148 section outputs a 3-bit binary encoded value that reflects the highest priority request line which is currently active. The encoder outputs, in turn, are applied to the inputs of the 3205 decoder. As a result, the 3205 section activates one of its eight bus priority in (BPRN/) outputs; that is, the BPRN/ line which is associated with the highest priority active request. BREQ7/ has highest priority; BREQØ/ has lowest priority.

An additional bus request input (BREQ8/) and bus priority in output (BPRN8/) are also provided. When active, BREQ8 disables the 3205 decoder, thus preventing any other BPRN line from being active. In the absence of a low level on pin P2-27, the BPRN8/ line will always remain active; thus granting the module attached to line BPRN8/ priority over all other modules.

All bus request lines, except BREQØ/, reach the module via the P2 auxiliary connector. All bus priority outputs, except BPRNØ/, exit the module via connector P2. BREQØ/ and BPRNØ/, which are attached to the Central Processor Module (the lowest priority master), communicate via the P1 bus connector.

The bus priority logic also includes provisions for generating the bus clock signals, BCLK and CCLK.

pp 4-05 through 5-14
missing from original

The CRT USART accepts one input from the CRT device, data set ready (CRT DSR/). The CRT USART outputs one line to the CRT device, data terminal ready (CRT DTR/).

Interrupts

Like most of the other I/O interfaces on the Monitor Module, the CRT interface includes provisions that allow it to be interrupt driven. When the USART has a data byte for the CPU, the Rx RDY output (pin 14 on the USART) goes true. Rx RDY clocks a 7474 flip-flop set. The Q output (CRT INP INT/) goes low. CRT INP INT/ constitutes bit 5 in the interrupt status word (see Section 5.2.10). CRT INP INT/ indicates that the CRT interface requires service to input a data byte to the CPU. When the CRT USART is ready to receive a data byte from the CPU, the Tx RDY output (pin 15) goes true, clocking another 7474 flip-flop set. The $\overline{Q}$ output of this flip-flop (CRT OUT INT/) constitutes bit 0 of the interrupt status word. CRT OUT INT/ indicates that the CRT USART requires service to receive the next data byte from the CPU. If jumper 7-8 is disconnected and jumper 8-9 is connected, CRT OUT INT/ will be generated as a result of the Tx EMPTY output (pin 18), instead of Tx RDY.

Both the CRT INP INT/ and CRT OUT INT/ flip-flops can be reset as the result of an interrupt control output sequence (see Section 5.2.10).

5.2.6 TELETYPE (TTY) INTERFACE

The TTY interface has been implemented with an 8251 USART chip (A17), as shown on sheet 2 of the module schematic, Figure 5-18.

Like the CRT USART, the TTY presents a parallel, 8-bit interface to the system data bus and a serial interface to the TTY device. The two "sides" of the TTY interface are described separately.

Parallel Interface

The TTY parallel interface is enabled by the TTY EN/ signal, which, you recall, is generated in the I/O command decode logic (Section 5.2.2) whenever the CPU executes an I/O instruction directed to port $F5_{16}$ or $F4_{16}$. The accompanying I/O command, I/O read (IORC/) or I/O write (IOWC/), dictates the direction of data flow. The least significant address bit, ADRØ/, differentiates between port addresses $F5_{16}$ (ADRØ=1) and $F4_{16}$ (ADRØ=0).

Control Output

An output instruction (IOWC/ is true) to port $F5_{16}$ (TTY EN/ and ADRØ/ are true) causes the TTY USART to accept a control byte through its data bus pins (DBØ–DB7). The control byte can be either a Mode Control Word or a Command Control Word. The TTY USART distinguishes between the two according to the same algorithm that we defined for the CRT USART (see Figure 5-6). Table 5-3 also describes bit definitions for the TTY Mode Control Word, with the single exception being that the TTY USART can only operate at 110 baud. Normally, the TTY Mode Control Word is equal to CD_{16}, specifying an 11 level code (eight data bits, disabled parity and two stop bits). The TTY Command Control Word is exactly as defined for the CRT in Figure 5-7.

Data Output

An output instruction to port $F4_{16}$ (TTY EN/ is true and ADRØ/ is false) causes the TTY USART to accept a data byte through its data bus pins. Bit 0 is the least significant bit and bit 7 is the most significant bit. The TTY interface will subsequently transmit the data byte, in serial fashion, to the TTY device as described in a later paragraph.

Status Input

An input instruction (IORC/) to port $F5_{16}$ (TTY EN/ and ADRØ/ are true) causes the TTY USART to output a status byte from its data bus pins. The status byte is enabled through the bidirectional data bus driver logic (see Section 5.2.11) and out onto the system data bus (pins P1-67 through 74). The status bits are the result of status and error checking functions performed within the USART. Bit definitions for the status byte are given in Figure 5-9.

Data Input

An input instruction to port $F4_{16}$ (TTY EN/ is true and ADRØ/ is false) causes the TTY USART to output a data byte (previously received from the

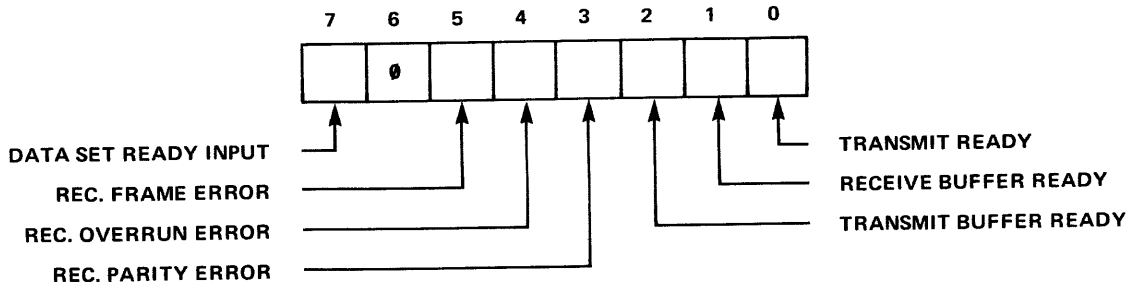


Figure 5-9. Bit Definitions for TTY Status Word

TTY device, as described below) from its data bus pins. The data byte is enabled through the bidirectional data bus. Bit 0 is the least significant bit and bit 7 is the most significant or parity bit from the TTY.

Timing for the USART's parallel interface to the system data bus is provided by the USART CLK signal (see Section 5.2.4). The USART can be reset by a high pulse on the SYS RST line.

Serial Interface

The TTY USART presents a serial, asynchronous, current loop interface to the TTY device. The USART baud rate (110 baud) is derived from a 1760-Hz timing signal received through the USART's TxC and RxC inputs (1760 Hz÷16 cycles/bit=110 baud). The timing signal is supplied by the USART clock generator logic (see Section 5.2.4).

When the TTY USART has a data bit for the device (previously received in parallel from the data bus), it serially transmits the data bit over the TTY Tx DATA line (pin J1-4). Notice that the USART's clear to send data (CTS/) output is tied to the USART's request to send data (RTS/) input. The data bit from the USART's TxD output (pin 19) is inverted and applied to the base of transistor Q5. Q5 turns on, allowing the +5V source (shown on the schematic) to drive current through the transistor and the TTY Tx DATA line.

The TTY device sends data serially to the TTY interface via the TTY Rx DAT line (pin J1-16). The device will cause an open circuit on the TTY Rx DAT line when it is sending a logical 0. Transistor Q3 turns off; consequently, a +5V source (tied to the collector of Q3) is applied to a 7404 inverter and a low voltage level appears at pin 3 on the USART. Notice that the first logical 0 bit received, the start bit, resets the latch that stores advance tape commands to the TTY paper tape reader (TTY ADV RDR/). TTY ADV RDR/ is described later in this sub-section.

When the TTY device sends a logical 1 bit, the receive line is essentially shorted to TTY Rx DAT RET (47Ω to +12 VDC). This activates transistor Q3, opening a path between the +5V source (tied to the collector) and ground. The resultant low voltage level is inverted by the 7404 section and a high voltage level appears at pin 3 on the USART.

The TTY USART accepts one input from the TTY device, data set ready (TTY DSR/). The TTY interface outputs one control command, the TTY reader control signal (TTY RDR CTL), as we mentioned above. TTY RDR CTL is generated when the CPU executes an output instruction to port $F9_{16}$ and the data byte that is output has a logical 1 in bit position one. Recall from Section 5.2.2, that the occurrence of IOWC/ while a value of $F9_{16}$ is present on the address lines results in the generation of the paper tape control signal (PT CTL/). PT CTL/ is inverted and NANDed (gate A38-11,12,13 on sheet 2 of the module schematic) together with data bit 1 (DAT1) to produce TTY ADV RDR/. TTY ADV RDR/ is applied to a S-R latch (consisting of two 7400 negative-OR gates). This latch stores the signal. On the trailing edge of TTY ADV RDR/, the contents of the latch are gated through to the base of transistor Q4. The transistor turns on and drives current through the TTY RDR CTL line (pin J1-12) to the TTY paper tape reader. After the tape is advanced one character and the start bit (logical 0) appears on the receive data line

(TTY Rx DAT), the low voltage level that is applied to pin 3 of the USART also is applied to one of the inputs of the top 7400 section in the latch, causing the latch to reset. Transistor Q4 turns off and TTY RDR CTL goes false. Figure 5-10 illustrates timing for TTY RDR CTL.

Interrupts

Like the CRT interface, the TTY interface includes provisions that allow it to be interrupt driven. When the USART has a data byte for the CPU, the Rx RDY output (pin 14 on the USART) goes true. Rx RDY clocks a 7474 flip-flop set. The $\overline{Q}$ output (TTY INP INT/) goes low. TTY INP INT/ constitutes bit 1 in the interrupt status word (see Section 5.2.10). TTY INP INT/ indicates that the TTY interface requires service to input a data byte to the CPU. When the TTY USART is ready to receive a data byte from the CPU, the Tx RDY output (pin 15) goes true, clocking another 7474 flip-flop set. The Q output of this flip-flop (TTY OUT INT/) constitutes bit 0 of the interrupt status word. TTY OUT INT/ indicates that the TTY USART requires service to receive the next data byte from the CPU. If jumper 11-12 is disconnected and jumper 10-11 is connected, TTY OUT INT/ will be generated as a result of the Tx EMPTY output (pin 18), instead of Tx RDY.

Both the TTY INP INT/ and TTY OUT INT/ flip-flops can be reset as the result of an interrupt control output sequence (see Section 5.2.10).

5.2.7 HIGH-SPEED PAPER TAPE READER/ PUNCH INTERFACE

The high-speed paper tape reader/punch (HSPTR/P) interface consists of three 7474 D-type flip-flops, twenty-two 8097 bus rivers, two 3404 6-bit latches (actually only 8 of the 12 bits are used), and various gating circuits, as shown on sheets 2 and 3 of the module schematic, Figure 5-18.

Recall from Section 5.2.2, that the I/O command decode logic generates the following paper tape-related signals when the appropriate port address

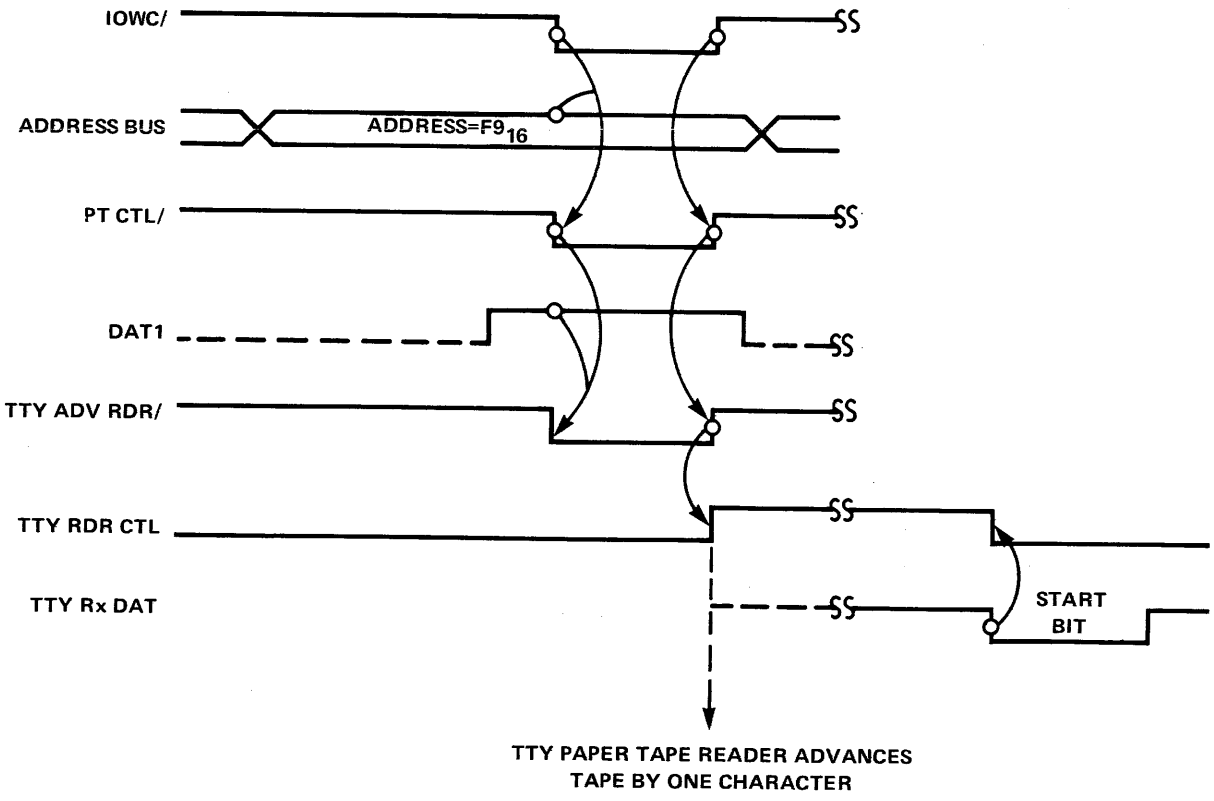


Figure 5-10. Timing for TTY Reader Control

and I/O read or write signal is received by the Monitor Module:

COMMAND SIGNAL		PORT ADDRESS (HEX)		I/O COMMAND
PTP DAT/	=	F8	AND	IOWC/ (output)
PT CTL/	=	F9	AND	IOWC/ (output)
PTR DATA/	=	F8	AND	IORC/ (input)
PT STAT/	=	F9	AND	IORC/ (input)

Data Output

When an output instruction to port $F8_{16}$ is executed (PTP DAT/ is true), the data byte, that is placed on the system data bus by the CPU, is received by the Monitor Module (at pins P1-67 through 74) and fed to the module's bidirectional data bus driver logic (see Section 5.2.11). The driver logic inverts the data bits and applies them to the eight inputs of the 3404 latches. The eight data bits are inverted by the 3404 section and output to 8097 non-inverting driver circuits which make the data byte available to the paper tape punch device (via connector J1).

Control Output

When an output instruction to port $F9_{16}$ is executed (PT CTL/ is true), the control byte that is placed on the system data bus by the CPU is received by the Monitor Module and gated together with PT CTL/ to produce the appropriate command signal(s) for the high-speed paper tape reader/punch or the TTY paper tape reader (see Section 5.2.6). Figure 5-11 gives bit definitions for the paper tape control byte.

If data bit 3 (DAT3) is true, the resultant output of NANDing DAT3 and PT CTL pre-sets a 7474 flip-flop. The Q output of this flip-flop enables a forward or reverse advance of tape on the high-speed reader as determined by bit 2 (DAT2) of the control byte. If DAT2 is true (logical 1), PTR DRV LFT/ (pin J1-17) causes the high-speed paper tape reader to advance the tape one character in the reverse (left) direction. PTR DRV RT/ (pin J1-22) causes the reader to advance the tape one character in the forward (right) direction. When the reader completes the tape advance (in either direction), the paper tape reader ready signal (PTR RDY/) goes true and pre-resets the enabling 7474 flip-flop.

If data bit 5 (DAT5) of the control byte is true (logical 1), the resultant output of NANDing DAT5 and PT CTL produces the paper tape advance signal (PTP ADV). If data bit 4 (DAT4) is false (logical 0), PT CTL/ enables DAT4 through a 3404 inverter; the inverted output (PTP FOR) is driven to the punch by an 8097 circuit. The occurrence of a true level on PTP FOR (pin J1-63) will cause the punch to advance one character in the forward direction on the positive-going edge of PTP ADV (pin J1-60).

If data bit 1 (DAT1) of the control byte is true (logical 1), the resultant output of NANDing DAT1 and PT CTL will generate TTY ADV RDR/. TTY ADV RDR/ causes the TTY paper tape reader to advance one character as described in Section 5.2.6.

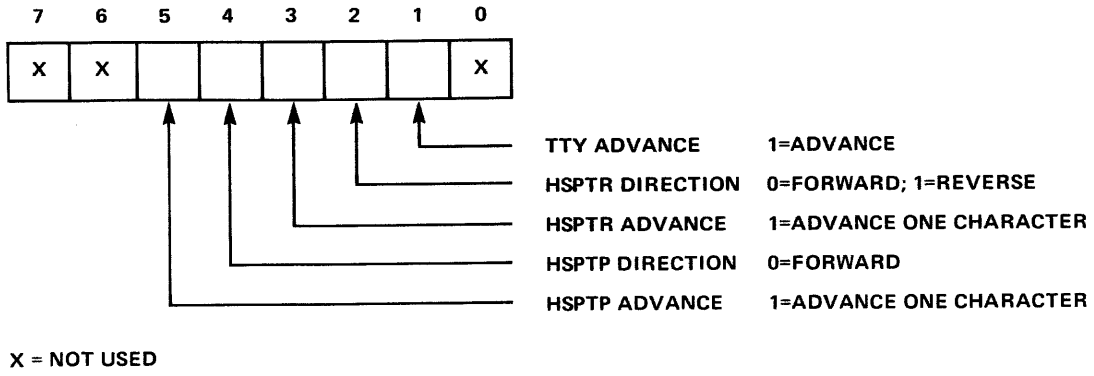


Figure 5-11. Bit Definitions for Paper Tape Control Byte

Data Input

When an input instruction to port $F8_{16}$ is executed (PTR DATA/ is true), the data byte from the high-speed paper tape reader (received at pins J1-64 to 71) is enabled through eight 8097 circuits by PTR DAT/ and placed on the system data bus (DATØ/–DAT7/) at pins P1-67 through 74.

Status Input

When an input instruction to port $F9_{16}$ is executed (PT STAT/ is true), the six status bits from the high-speed paper tape reader and punch (received at pins J1-21, 72, 74, 78, 80, 82) are enabled through six 8097 circuits by PT STAT/ and placed on the system data bus (DATØ–DAT5) at pins P1-69 through 74. Figure 5-12 gives bit definitions for the status byte.

Interrupts

The paper tape reader/punch interface includes provisions that allow it to be interrupt driven. A high-to-low transition on the PTP RDY/ or PTR RDY/ status line clocks the 7474 flip-flop associated with that status line to the set state. The $\overline{Q}$ outputs of these two 7474 sections constitute the punch (PTP OUT INT/) and reader (PTR INP INT/) interrupt status lines to the Monitor interrupt logic (see Section 5.2.10). PTP OUT INT/ is bit 2 and PTR INP INT/ is bit 3 of the interrupt status word. Both the PTP OUT INT/ and PTR INP INT/ flip-flops can be reset as the result of an interrupt control output sequence, as described in Section 5.2.10).

5.2.8 PROM PROGRAMMER INTERFACE

The PROM Programmer interface consists of eight 8097 bus receiver/driver circuits and various gating circuits, as shown on sheets 2 and 3 of the module schematic, Figure 5-18.

Recall from Section 5.2.2, that the I/O command decode logic generates the following PROM-related signals when the appropriate port address and I/O read or write signal is received by the Monitor Module:

COMMAND SIGNAL		PORT ADDRESS (HEX)		I/O COMMAND
PROM ADR HIGH-CTL	=	F1	AND	IOWC/
PROM ADR LOW	=	F2	AND	IOWC/
PROM WRT DATA	=	FØ	AND	IOWC/
EN PROM RD DATA	=	FØ or F1	AND	IORC/
PROM RD DATA	=	FØ	AND	IORC/
PROM RD STATUS	=	F1	AND	IORC/

High Address – Control Output

When an output instruction to port $F1_{16}$ is executed (PROM ADR HIGH-CTL is true), the PROM address-control byte, that has been output to the system data bus by the CPU, is received by the Monitor Module (at pins P1-67 through 74) and fed to the module's bidirectional data bus driver logic (see Section 5.2.11) which asserts the byte on the peripheral data out bus (via connector J1). The resultant output of NANDing command strobe

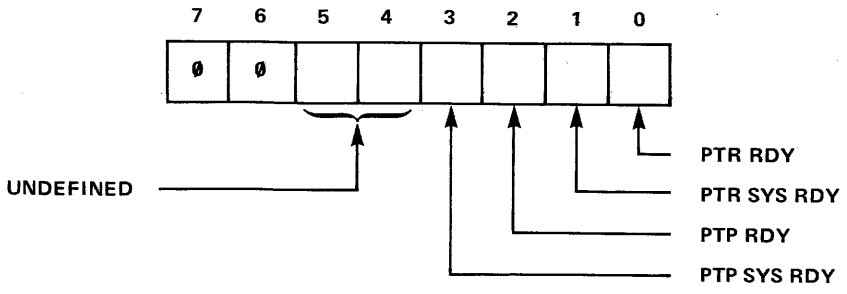


Figure 5-12. Bit Definitions for Paper Tape Status Word

(CMND STRB – see Section 5.2.3) and PROM ADR HIGH-CTL provides a PROM control pulse (PROM CTL PLS//) to the PROM Programmer peripheral (via pin J1-98). The address-control byte consists of the four most significant address bits (on data lines 0, 1, 2, and 3) and four control bits (on data lines 4, 5, 6, and 7).

Low-Address Output

When an input instruction to port $F2_{16}$ is executed (PROM ADR LOW is true), the PROM low-address byte (not to be confused with the I/O port address bytes on the address bus), that has been output to the system data bus by the CPU, is received by the Monitor Module (at pins P1-67 through 74) and fed to the module's bidirectional data bus driver logic which asserts the byte on the peripheral data out bus (via connector J1). The resultant output of NANDing command strobe (CMND STRB) and PROM ADR LOW provides a PROM address pulse (PROM ADR PLS/) to the PROM Programmer (via pin J1-100). The PROM low-address byte constitutes the eight least significant address bits (bit 0 is the LSB) of the 12-bit PROM address that is formed by concatenating bits 0–3 of the address-control byte described in the previous paragraph with the PROM low-address byte.

Data Output

When an output instruction to port $F\emptyset_{16}$ is executed (PROM WRT DATA is true), the data byte, that has been output to the system data bus by the CPU, is received by the Monitor Module and fed to the module's bidirectional data bus driver logic, which asserts the byte on the peripheral data out bus (via connector J1). The resultant output of NANDing command strobe (CMND STRB) with PROM WRT DATA provides a data pulse (PROM WRT DAT PLS/) to the PROM Programmer (via pin J1-96). The data byte is written into the PROM location specified by the 12-bit PROM address described above.

Data or Status Input

An input instruction addressed to port $F\emptyset_{16}$ or $F1_{16}$ (EN PROM RD DATA/ is true) specifies that a data or status byte is to be input from the PROM Programmer to the CPU. If port $F\emptyset_{16}$ is addressed, the PROM RD DATA signal is applied to an 8098 inverting driver and output to the PROM Programmer from pin J1-89 as PROM RD DAT/.

If port $F1_{16}$ is addressed, the PROM RD STATUS signal is driven by an 8098 circuit through pin J1-90 to the PROM Programmer under the mnemonic PROM RD STAT/. The data or status byte that is input (at pins J1-85 through 88 and 91 through 94) to the Monitor Module is enabled through eight 8097 receiver/driver circuits by EN PROM RD DATA/. The 8097 circuits, in turn, drive the data or status byte onto the system data bus (through pins P1-67 to 74).

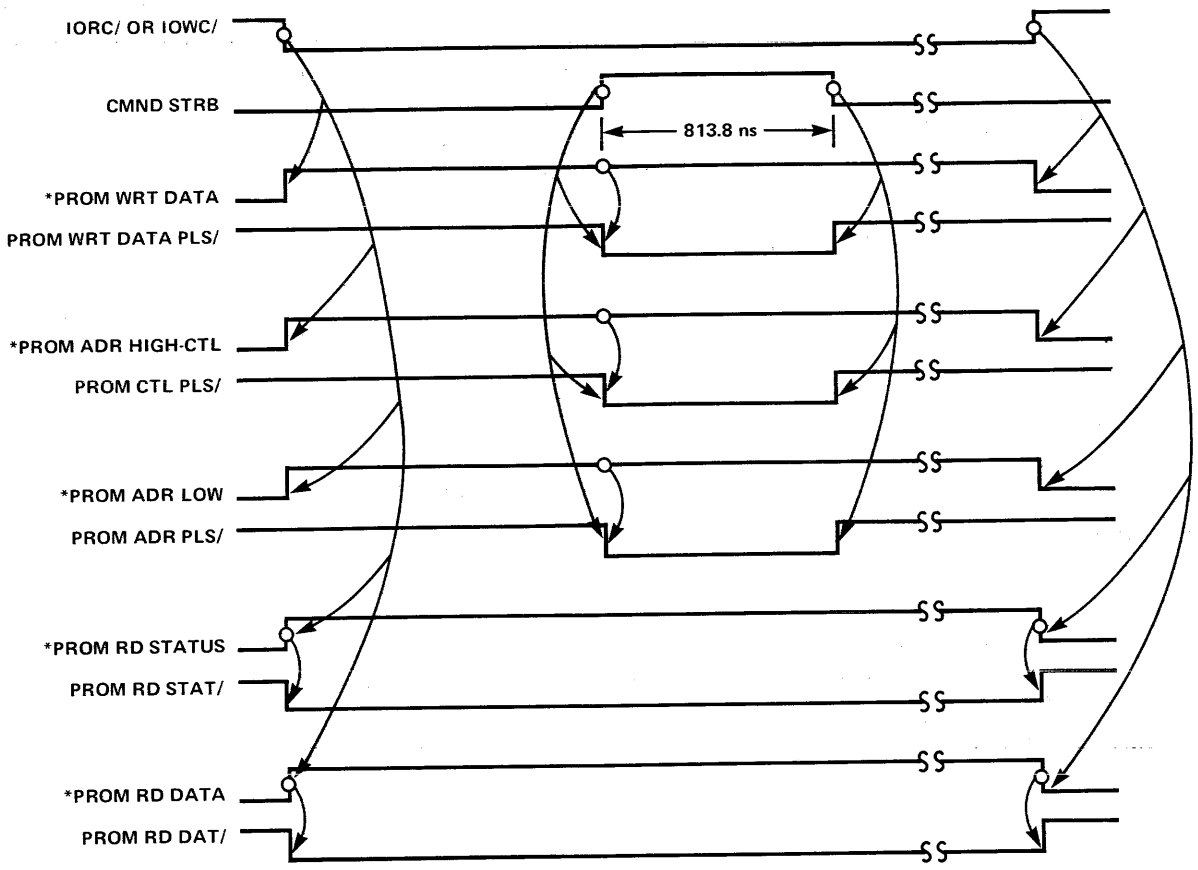
Timing for the PROM Programmer interface is illustrated in Figure 5-13.

5.2.9 LINE PRINTER INTERFACE

The line printer interface has been designed to operate with printers which are capable of receiving input commands as coded ASCII characters in the same manner as data. Timing inputs must be provided by the printer electronics and sensed by the Monitor Module on the two status inputs. The line printer interface consists of a 7474 flip-flop (which latches LPT ACK/) and various gating circuits which are used to generate two control outputs and a data strobe, as well as the circuits that receive the two status inputs and place them onto the system data bus. All of the line printer interface logic is shown on sheet 2 of the module schematic, Figure 5-18.

Recall from Section 5.2.2, that the I/O command decode logic generates the following line printer-related signals when the appropriate port address and I/O read or write signal is received by the Monitor Module:

COMMAND SIGNAL		PORT ADDRESS (HEX)		I/O COMMAND
LPT DAT/	=	FA	AND	IOWC/
LPT CTL/	=	FB	AND	IOWC/
LPT STAT/	=	FB	AND	IORC/



*NOTE: ONLY ONE OF THESE SIGNALS WILL BE TRUE IN ANY GIVEN CYCLE.

Figure 5-13. PROM Programmer Interface Timing

Data Output

When an output instruction to port FA_{16} is executed, LPT DAT is NANDed with CMND STRB (see Section 5.2.3) to generate the line printer data strobe (LPT DAT STRB/) which is available to the line printer at pin J1-95. Data output to the system data bus by the CPU is received by the Monitor Module (at pins P1-67 through 74), gated through the bidirectional data bus driver logic (Section 5.2.11) and made available on the peripheral data out bus (via connector J1).

Control Output

When an output instruction to port FB_{16} is executed, LPT CTL is NANDed separately with bits 0 (DATØ) and 1 (DAT1) of the control byte (that was output by the CPU) to form two control outputs to the line printer device, LPT CTLØ/ (pin J1-24) and LPT CTL1/ (pin J1-23).

Status Input

When an input instruction to port FB_{16} is executed, LPT STAT/ enables two 8097 non-inverting drivers which pass the two status bits from the line printer, LPT BUSY (pin J1-56) and LPT STAT1/ (pin J1-54), to data lines 0 and 1 of the system data bus (DATØ/ and DAT1/).

Interrupts

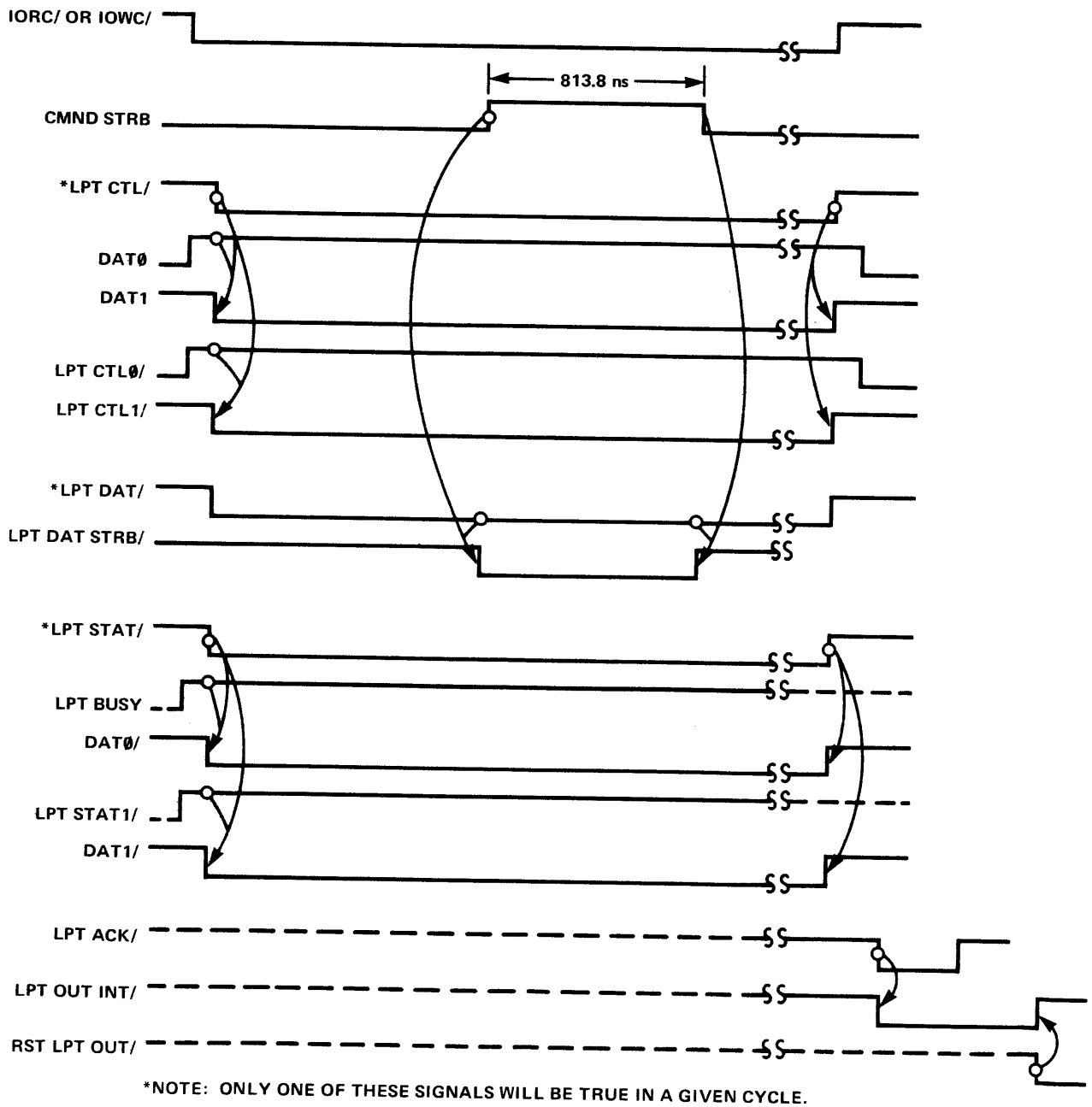
When the line printer device accepts a data byte from the system data bus, it returns an acknowledge signal (LPT ACK/) to the interface (at pin J1-25). On its positive-going edge, LPT ACK/ clocks a 7474 latch set. The $\overline{Q}$ output of the 7474 section (LPT OUT INT/) constitutes bit 6 of the interrupt status word. The LPT OUT INT/ flip-flop can be reset as the result of an interrupt control output sequence, as described in Section 5.2.10.

Timing for the line printer interface is illustrated in Figure 5-14.

5.2.10 MONITOR INTERRUPT LOGIC

The Monitor interrupt logic groups the seven interrupt lines from the TTY, CRT, paper tape reader/punch, and line printer interfaces into a status word that can be read, under program control, by the CPU. In addition, the interrupt logic will, if enabled, issue an interrupt request on level 3 when one of the interface interrupt lines goes true. The monitor interrupt logic consists of six 8097 bus drivers, one 8093 bus driver, one 7474 D-type flip-flop, and assorted gating circuits, as shown on sheet 3 of the module schematic, Figure 5-18.

Recall from Section 5.2.2, that the I/O command decode logic generates the following interrupt-related signals when the appropriate port and I/O



*NOTE: ONLY ONE OF THESE SIGNALS WILL BE TRUE IN A GIVEN CYCLE.

Figure 5-14. Line Printer Interface Timing

read or write signal is received by the Monitor Module:

COMMAND SIGNAL		PORT ADDRESS (HEX)		I/O COMMAND
INT CTL/	=	F3	AND	IOWC/
INT STAT/	=	FA	AND	IORC/

Control Output

When an output instruction to port $F3_{16}$ is executed (INT CTL/ is true), the control byte that is output to the system data bus by the CPU is received by the Monitor (at pins P1-67 through 74), passed through the bidirectional data bus driver logic (see Section 5.2.11), and supplied to the interrupt logic. Bits 0–6 of the control byte specify whether a particular interface interrupt flip-flop is to be reset (e.g., if data bit 0=logical 1, the TTY OUT INT flip-flop is reset). Bit 7 enables or disables the interrupt logic's ability to assert an interrupt request on level 3. Specific bit definitions for the interrupt control byte are given in Figure 5-15.

Bits 0–6 of the control byte are each applied to one of two inputs on seven 7400 NAND gates. The other input to these 7400 gates is supplied by the ORed result of INT CTL and SYS RST (the system reset pulse). Consequently, during an interrupt control output sequence, those 7400 gates that are associated with a data line that represents a logical 1 bit are activated. The output of each active 7400 gate constitutes one of seven reset interrupt signals, as defined below:

DATA BIT	RESET INTERRUPT SIGNAL
0	RST TTY OUT/
1	RST TTY INP/
2	RST PTP OUT/
3	RST PTR INP/
4	RST CRT OUT/
5	RST CRT INP/
6	RST LPT OUT/

These clear signals reset the appropriate interrupt flip-flop in their associated interface.

Bit 7 of the control byte is applied to the D input of a 7474 flip-flop in the interrupt logic. The positive-going edge of INT CTL/ clocks the level on the D-input into the latch. The Q output of this enable-interrupt flip-flop feeds one input of a 7426 NAND gate. The other input is supplied by a 7430 eight-input OR gate. When any one of the interrupt lines from the I/O interfaces (e.g., TTY OUT INT/) is true, this 7430 provides a high level to the 7426 input. If the enable-interrupt latch is set, the open collector output from the 7426 gate drives an active-low interrupt request on the interrupt priority level 3 line (INT3/; pin P1-40).

A true level on the system reset line (SYS RST/) resets the enable interrupt flip-flop and causes all seven reset interrupt signals to the interfaces to be generated.

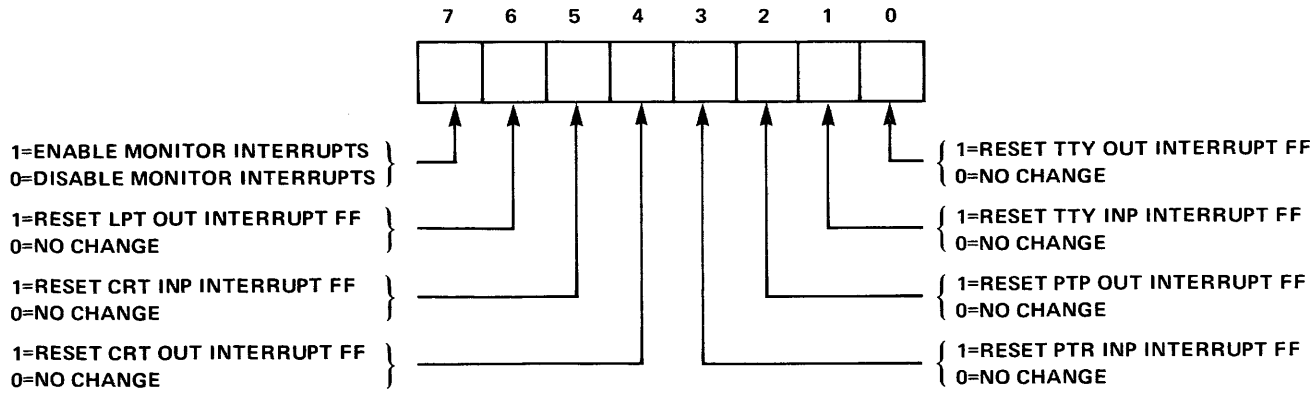


Figure 5-15. Bit Definition for Interrupt Control Word

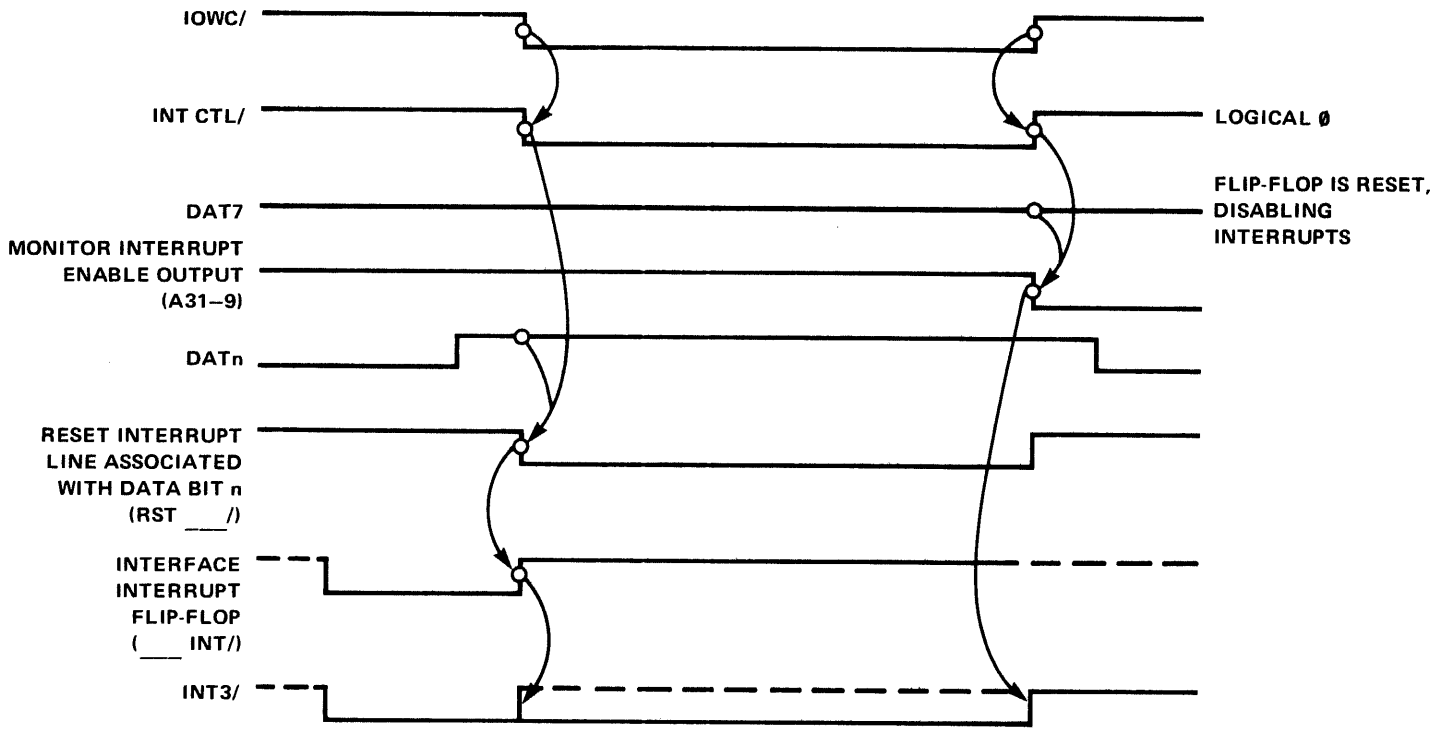


Figure 5-16. Interrupt Control Logic Timing

Timing for the interrupt control logic is illustrated in Figure 5-16.

Status Input

When an input instruction to port FA_{16} is executed, INT STAT/ enables the seven I/O interface interrupt lines through 8097 bus driver circuits (LPT OUT INT/ is actually driven by an 8093 circuit) and out onto the system data bus (DATØ/–DAT6/). The levels on these interrupt lines constitute the interrupt status word, shown in Figure 5-17. This status word could be interrogated by a user-generated operating system or application routine to determine the exact source(s) of interrupt request(s), resolve priority among concurrent requests, and call the proper I/O service routine.

5.2.11 BIDIRECTIONAL DATA BUS DRIVER LOGIC

The bidirectional data bus driver logic routes status or data input bytes from the TTY or CRT USART's onto the system data bus. In the other direction, the driver logic routes data output or control bytes from the system data bus to the TTY or CRT data bus pins, the paper tape punch output bus, the peripheral data out bus, or the Monitor interrupt logic. The driver logic consists of two 8226 4-bit bidirectional bus drivers and other assorted gates, inverters and drivers, as shown on sheet 3 of the module schematic, Figure 5-18.

When an input instruction specifying that a data or status byte is to be input from the TTY or CRT USART, the I/O command decode logic (section 5.2.2) generates the CRT OR TTY/ signal. CRT OR TTY/ is NANDed with IORC/ (the I/O read signal from the CPU) at a 7432 section. The active-low output from this gate is applied to the direction control enable (DCE) inputs on the 8226 bus drivers. A low level at the DCE inputs directs the 8226 devices to gate data or status from the USART data bus lines to the system data bus, DATØ/–DAT7/ (pins P1-67 to 74).

Except during TTY or CRT input cycles, the level on the 8226's DCE inputs is high, reversing the direction of data flow through the 8226 devices. Output data, placed on the system data bus by the CPU (or another bus master module), enters the monitor module at pins P1-67 to 74 and flows through the two 8226 drivers to the TTY or CRT USART data bus pins, the paper tape punch output bus logic, the peripheral data out bus or the Monitor interrupt logic.

If the data or control byte is intended for the TTY or CRT USART's, the TTY EN/ or CRT EN/ and the IOWC/ signals enable the byte into the appropriate USART, as described in Sections 5.2.5 and 5.2.6.

If the data byte is intended for the paper tape punch, the PTP DAT/ signal enables the byte through the 3404 latches in the PTP output bus logic and out to punch device (via the J1 connector).

If the data, control or address byte is intended for the PROM Programmer or line printer peripherals, the byte is driven onto the peripheral data bus by various 8098 and 8097 circuits (through connector J1). If the line printer is the specified destination, the LPT DAT STRB/ signal (see Section 5.2.9)

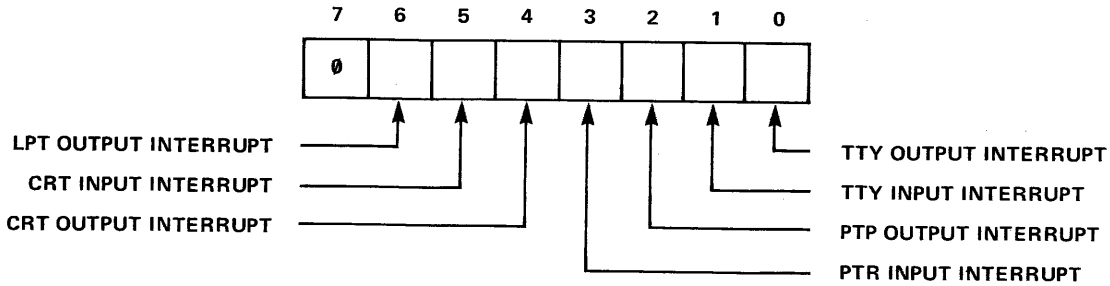


Figure 5-17. Bit Definitions for Interrupt Status Word

directs the line printer to accept the data byte. If the PROM Programmer peripheral is the specified destination, the PROM WRT DAT PLS/, PROM CTL PLS/ or PROM ADR PLS/ signal will direct the PROM Programmer to accept the data, address/control or address byte (see Section 5.2.8).

If the control byte is intended for the Monitor interrupt logic, the INT CTL/ signal will enable the control bits to reset the specified interrupt flip-flops, as described in Section 5.2.10.

5.2.12 MONITOR MODULE SCHEMATIC

Figure 5-18 provides a complete schematic drawing (3 sheets) of all logic on the Monitor Module

5.3 UTILIZATION: MONITOR MODULE

This section provides information on utilization of the Monitor Module.

5.3.1 INSTALLATION

In installing the Monitor Module, the user must take account of:

(a) environmental extremes
(b) mounting considerations
(c) electrical connections
(d) power requirements
(e) signal requirements
(f) jumper connections

Environment

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conducive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity should not exceed 90%, non-condensing.

Mounting

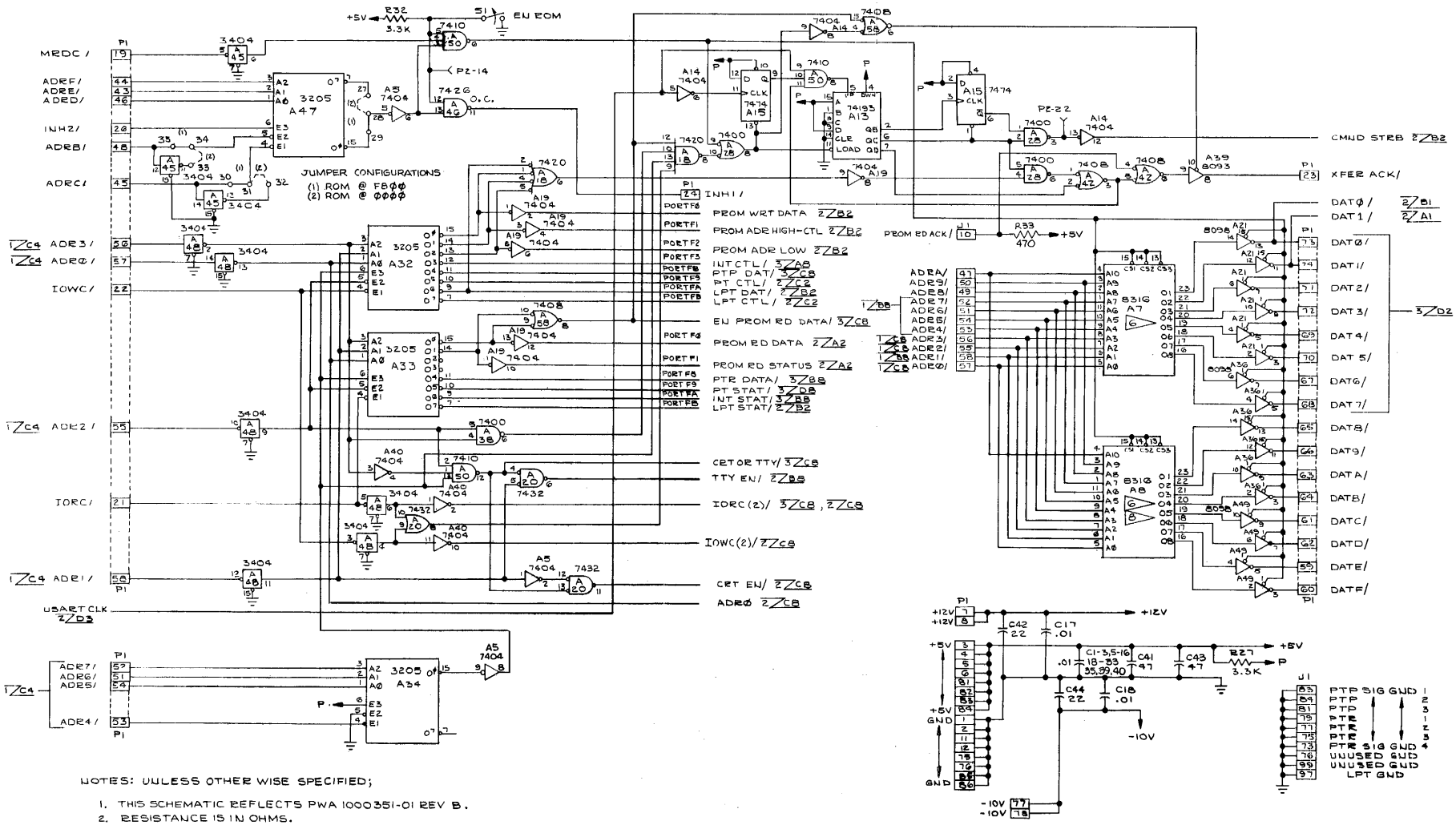
Avoid locating the module near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections on the board, resulting in abnormally high noise levels or outright failure of the assembly.

Dimensions of the module are 12-in. X 6.75-in. Be sure to allow enough additional clearance to ensure adequate cooling.

The module is designed to plug directly into three standard, double-sided PC edge connectors. An 86-pin connector and a 60-pin auxiliary connector are on one edge of the board; a 100-pin connector is on the opposite edge. The connectors can serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the assembly be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the board, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the components on the module.

Electrical Connections

The Monitor Module communicates with the motherboard and, consequently, the rest of the system, through a standard 86-pin, double-sided PC edge connector (P1), 0.156-in. contact centers, as shown in Figure 5-19. Control Data VPB01E43A-00A1 is one suitable type of connector. Pin allocations on this connector are given in Table 5-6 of Section 5.3.2. The module can also communicate with other modules in the system (or with test equipment), through the auxiliary 60-pin, double-sided PC edge connector (P2), 0.1-in. contact centers (see Figure 5-19). Pin allocations for this connector are listed in Table 5-7. The module transfers information to/from the peripheral devices via a 100-pin, double-sided PC edge connector (J1) which attaches to the edge opposite that of the other two connectors. This connector has 0.1-in. contact centers. Viking 3VH50/1JN5 is one suitable type of connector for communicating with the peripheral devices. Pin allocations for this connector are given in Table 5-8.



NOTES: UNLESS OTHER WISE SPECIFIED;

1. THIS SCHEMATIC REFLECTS PWA 1000351-01 REV B.
2. RESISTANCE IS IN OHMS.
3. CAPACITANCE IS IN MICRO-FARADS.
4. ALL DIODES ARE IN914.
5. USED ONLY WITH USART EMULATOR (C34,C36,C37, C38)
6. ROMS MUST BE ORDERED WITH INVERTED ADDRESS, POSITIVE DATA. (A7,A8)
7. A16, A17 (8251's) NOT USED WITH USART EMULATOR.
8. A8 IS USED FOR 16 BIT APPLICATION ONLY.
9. THESE PINS ARE NOT USED; J1-5,7,9,11,13,15,40

Figure 5-18. Monitor Module Schematic (Sheet 1 of 3)

intel © INTEL 1978	3065 BOWERS AVE. SANTA CLARA CALIF. 95051		
TITLE: SCHEMATIC MONITOR			
SIZE: D	DEPT: 410	DRAWING NO.: 1000353	REV: C

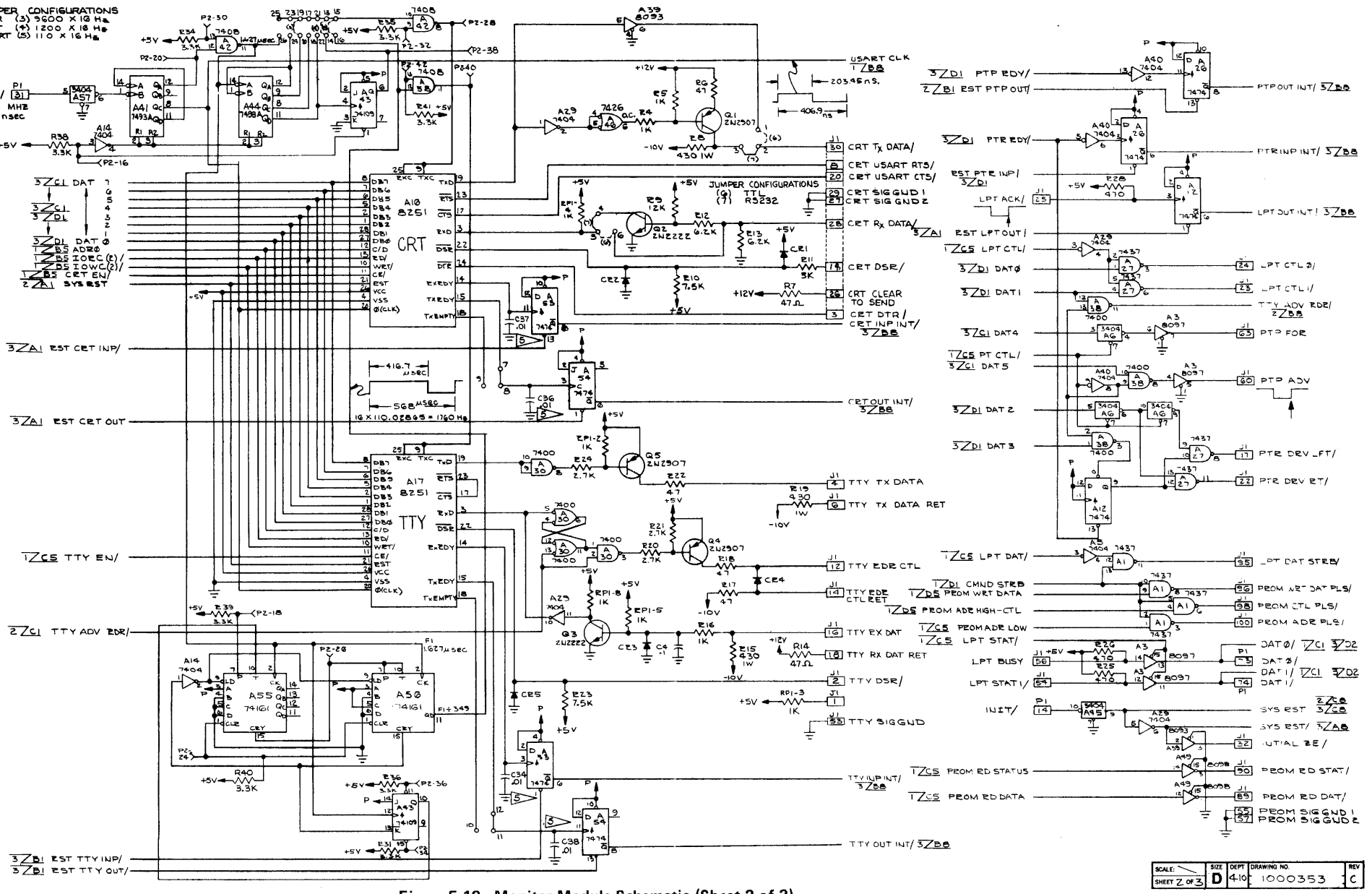


Figure 5-18. Monitor Module Schematic (Sheet 2 of 3)

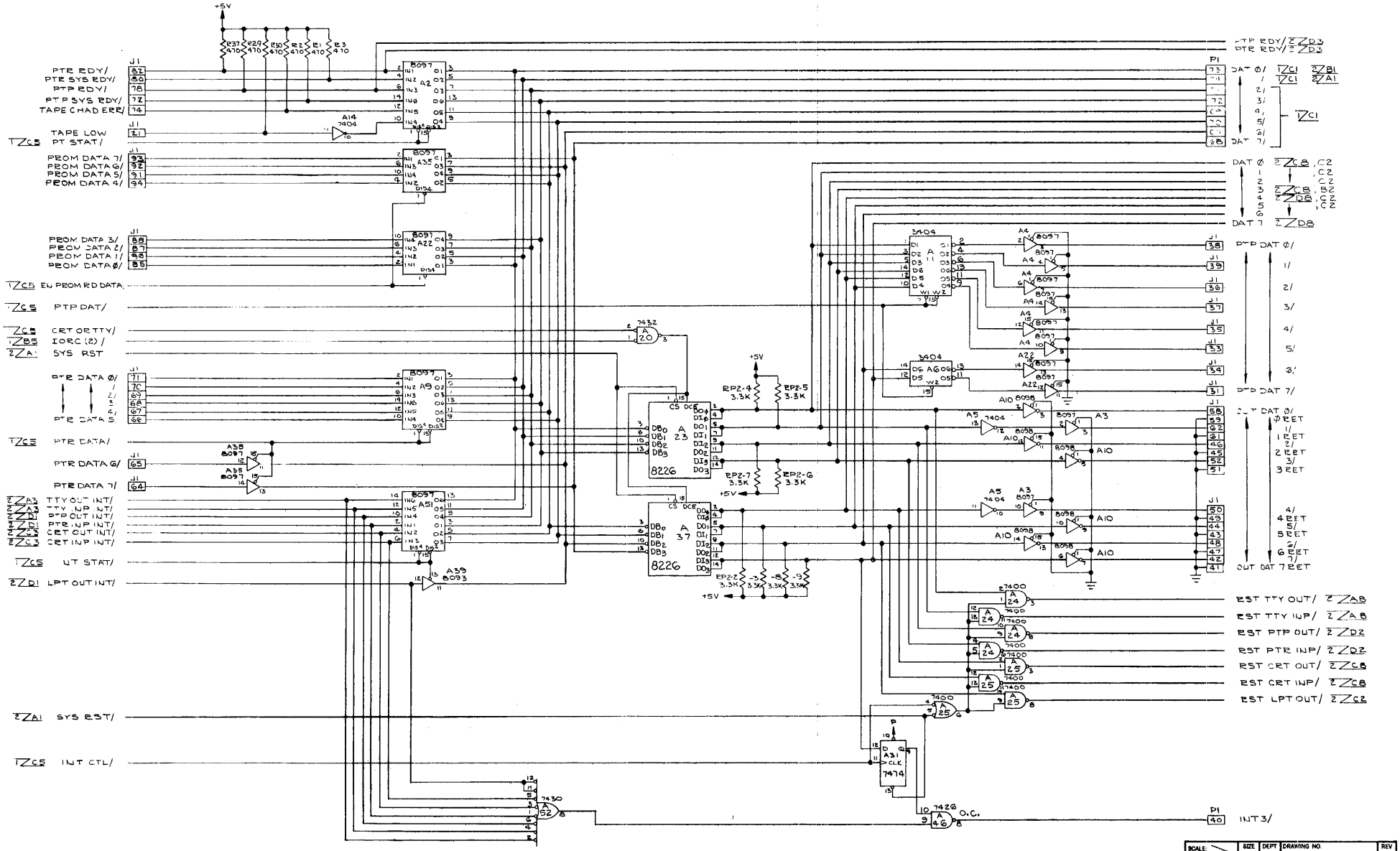


Figure 5-18. Monitor Module Schematic (Sheet 3 of 3)

SCALE:	SIZE	DEPT	DRAWING NO.	REV
SHEET 3 OF 3	D	410	1000353	C

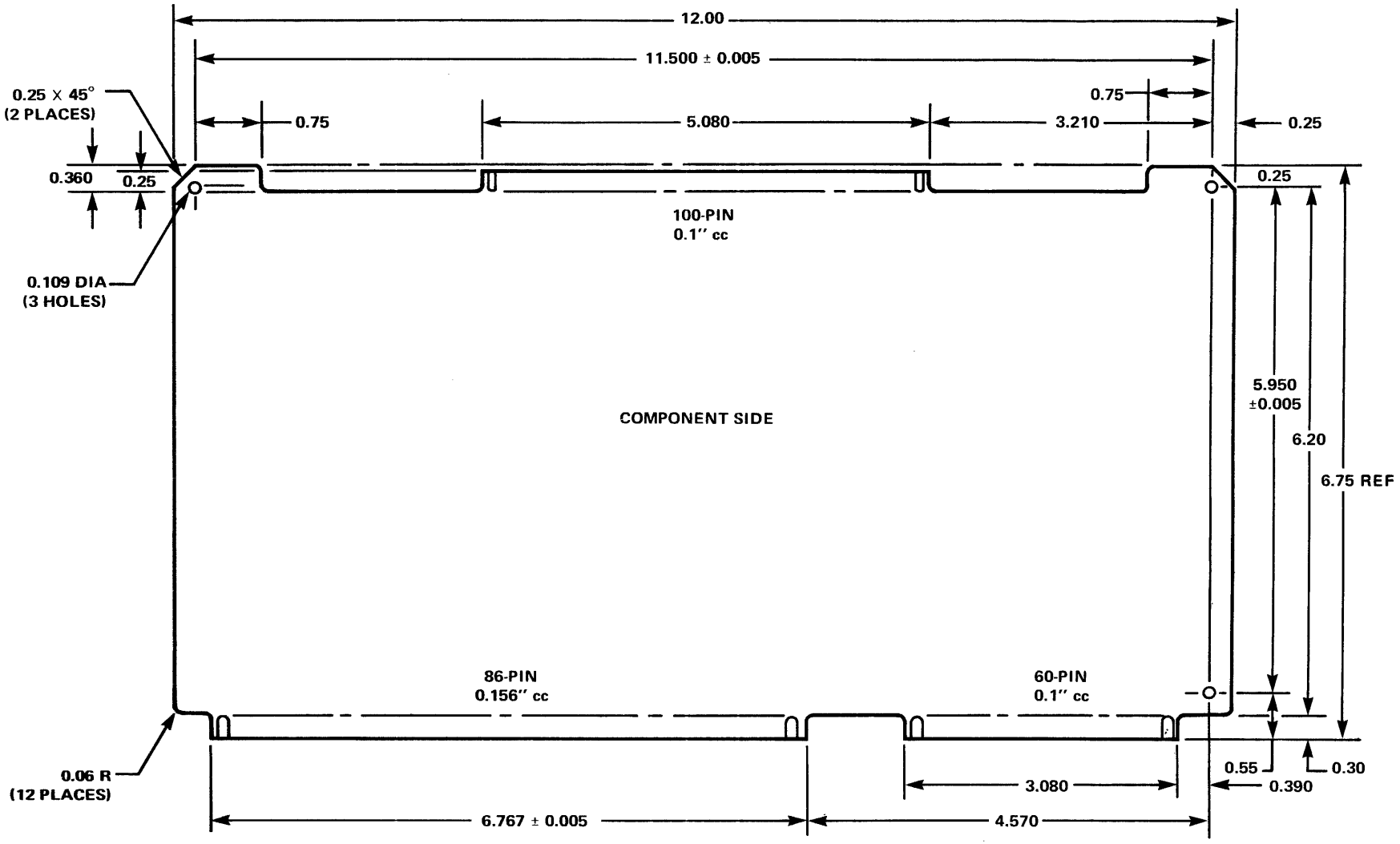


Figure 5-19. Monitor Module Connectors

The Monitor Module requires DC power at levels of +5 VDC, −10 VDC, and +12 VDC.

Refer to the pin lists in Tables 5-6 and 5-8 of Section 5.3.2 for power connectors.

Signal Requirements

All data and control functions appearing at the module edge connectors are at TTL levels (except those connected to TTY and CRT devices having RS232, current loop interfaces). Electrical characteristics of the signal inputs and outputs as well as power inputs are given in Section 5.4

Signal descriptions and connector pin allocations are given in Section 5.3.2

Jumper Connections

There are three groups of jumper pads on the Monitor Module.

The first group, consisting of jumper pads 1-2-3 and 4-5-6, is shown on sheet 2 of the module schematic, Figure 5-18. These two jumper pads determine the interface drive characteristics for the CRT interface. If jumper pads 2-3 and 4-5 are connected, the transmit (CRT Tx DATA/) and receive (CRT Rx DATA/) lines, respectively, will exhibit RS232 interface characteristics. If, however, jumper points 1-2 and 5-6 are connected instead, the transmit and receive lines, respectively, will exhibit TTL drive characteristics.

The second set of jumpers are also shown on sheet 2 of the module schematic. These seven jumper pairs provide different frequency timing signals for use in selecting a particular baud rate for data communication devices. Table 5-4 lists the seven jumper pairs with the frequency of the signals which they provide to the input of the 7408 gate (A42-10) shown on the schematic drawing. When enabled, the output of this 7408 gate is available at pin P2-28 (on the auxiliary connector).

The third group of jumper pads allows the Monitor ROM to respond to addresses other than those in the range $F8ØØ_{16}$–$FFFF_{16}$, as defined in Table 5-5. These jumper pads, 27-28-29, 33-34-35, and 30-31-32, are shown on sheet 1 of the module schematic.

Table 5-4

BAUD RATE JUMPER NETWORK CONNECTIONS

JUMPER PAIR	FREQUENCY (kHz)	BAUD RATE (baud)*
25-26	614.6	38,412
23-24	307.3	19,206
19-20	153.6	9,600
17-18	76.8	4,800
21-22	38.4	2,400
13-14	19.2	1,200
15-16	1.76	110

*Where baud rate = frequency ÷ 16 cycles/bit

NOTE: The MDS monitor utilizes a ÷ 16 for its TTY channel and a ÷ 64 for its CRT channel.

Table 5-5

ROM ADDRESS JUMPER NETWORK CONNECTIONS

JUMPER CONFIGURATIONS	ROM STARTING ADDRESS (HEX)
28-29, 30-31, 34-35 (standard)	F8ØØ
27-28, 30-31, 34-35	18ØØ
28-29, 31-32, 34-35	E8ØØ
27-28, 31-32, 34-35	Ø8ØØ
28-29, 30-31, 33-34	FØØØ
27-28, 30-31, 33-34	1ØØØ
28-29, 31-32, 33-34	EØØØ
27-28, 31-32, 33-34	ØØØØ

5.3.2 PIN LISTS: MONITOR MODULE

The following section provides connector pin allocations on the Monitor Module. The pins and their designated signal functions for the 86-pin connector (P1) are listed in Table 5-6. The same information for the 60-pin auxiliary connector (P2) is listed in Table 5-7. Pin and signal information for the 100-pin peripheral connector (J1) is given in Table 5-8.

Table 5-6

P1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	GND	Ground
2	GND	
3	+5 VDC	Power inputs
4	+5 VDC	
5	+5 VDC	
6	+5 VDC	
7	+12 VDC	Power inputs
8	+12 VDC	
9		
10		
11	GND	Ground
12	GND	
13		
14	INIT/	System reset
15		
16		
17		
18		
19	MRDC/	Memory read command
20		
21	IORC/	I/O read command
22	IOWC/	I/O write command
23	XACK/	Acknowledge transfer
24	INH1/	Inhibit RAM
25		
26	INH2/	Inhibit ROM
27		
28		
29		
30		
31	CCLK/	Bus clock (9.8304 MHz)
32		
33		
34		
35		
36		
37		
38		
39		
40	INT3/	Interrupt request on level 3
41		
42		
43	ADRE/	Address bus

PIN	SIGNAL	FUNCTION
44	ADRF/	Address bus
45	ADRC/	
46	ADRD/	
47	ADRA/	
48	ADRB/	
49	ADR8/	
50	ADR9/	
51	ADR6/	
52	ADR7/	
53	ADR4/	
54	ADR5/	
55	ADR2/	
56	ADR3/	
57	ADRØ/	
58	ADR1/	
59	DATE/	Data bus
60	DATF/	
61	DATC/	
62	DATD/	
63	DATA/	
64	DATB/	
65	DAT8/	
66	DAT9/	
67	DAT6/	
68	DAT7/	
69	DAT4/	
70	DAT5/	
71	DAT2/	
72	DAT3/	
73	DATØ/	
74	DAT1/	
75	GND	Ground
76	GND	
77	-10 VDC	Power inputs
78	-10 VDC	
79		
80		
81	+5 VDC	Power inputs
82	+5 VDC	
83	+5 VDC	
84	+5 VDC	
85	GND	Ground
86	GND	

Table 5-7

P2 CONNECTOR PIN LIST (Primarily Test Points)

PIN	SIGNAL	FUNCTION
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14	EN ROM	
15		
16	RST CNTRS/	
17		
18	÷349 P INPUT	
19		
20	1.627 μs	
21		
22	CMND STRB/	
23		
24	÷349 CLEAR/	
25		Test Points
26	÷32 OF ÷349	
27		
28	CRT BAUD CLK	
29		
30	DIS 1.627/	

PIN	SIGNAL	FUNCTION
31		
32	DIS CRT BAUD CLK	
33		
34	÷349 LSB RST/	
35		
36	÷349 LSB SET/	
37		
38	19.2 kHz	
39		
40	1760 Hz	
41		
42	DIS 1760 Hz	
43		
44		
45		
46		
47		
48		
49		
50	(NO CONNECTIONS)	
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		

Table 5-8

J1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	-----	+5 VDC Via 1K Resistor
2	TTY DSR/	TTY Data Set Ready (TTL)
3	CRT DTR/	CRT Data Terminal Ready (TTL)
4	TTY Tx DATA	TTY Transmit Data Line
5		
6	TTY Tx DATA RET	TTY Data Transmit Return
7		
8	CRT USART RTS/	CRT Request to Send Data (TTL)
9		
10	PROM RD ACK/	PROM Programmer Read Ack.
11		
12	TTY RDR CTL	TTY Reader Control
13		
14	TTY RDR CTL RET	TTY Reader Control Return
15		
16	TTY Rx DAT	TTY Receive Data Line
17	PTR DRV LFT/	PTR Advance Tape To Left
18	TTY Rx DAT RET	TTY Data Receive Return
19	CRT DSR/	CRT Data Set Ready; invert RS232
20	CRT USART CTS/	CRT Clear To Send Data (TTL)
21	TAPE LOW	PTP Status
22	PTR DRV RT/	PTR Advance Tape To Right
23	LPT CTL1/	LPT Control (not normally used)
24	LPT CTL0/	
25	LTP ACK/	LPT Data Acknowledge
26	CRT CLEAR TO SEND	CRT Clear To Send RS232
27	CRT SIG GND 2	Ground
28	CRT Rx DATA/	CRT Receive Data Line
29	CRT SIG GND 1	Ground
30	CRT Tx DATA	Transmit Data Line
31	PTP DAT 7/	PTP Data Output
32	INITIALIZE/	Initialize Pulse
33	PTP DAT 5/	
34	PTP DAT 6/	
35	PTP DAT 4/	
36	PTP DAT 2/	PTP Data Output
37	PTP DAT 3/	
38	PTP DAT Ø/	
39	PTP DAT 1/	
40		
41	OUT DAT 7 RET	
42	OUT DAT 7/	
43	OUT DAT 5 RET	
44	OUT DAT 5/	
45	OUT DAT 2 RET	Data Out Bus to Peripherals
46	OUT DAT 2/	
47	OUT DAT 6 RET	
48	OUT DAT 6/	
49	OUT DAT 4 RET	
50	OUT DAT 4/	

Table 5-8

J1 CONNECTOR PIN LIST (continued)

PIN	SIGNAL	FUNCTION
51	OUT DAT 3 RET	
52	OUT DAT 3/	
53	TTY SIG GND	Ground
54	LPT STAT 1/	LPT Status Bit
55	PROM SIG GND1	Ground
56	LPT BUSY	LPT Busy (Ready/)
57	PROM SIG GND2	Ground
58	OUT DAT 0/	Data Out Bus To Peripherals
59	OUT DATA 0 RET	
60	PTP ADV	PTP Advance Tape
61	OUT DAT 1 RET	Data Out Bus To Peripherals
62	OUT DAT 1/	
63	PTP FOR	Forward Tape Advance
64	PTR DATA 7/	
65	PTR DATA 6/	
66	PTR DATA 5/	
67	PTR DATA 4/	PTR Data Input
68	PTR DATA 3/	
69	PTR DATA 2/	
70	PTR DATA 1/	
71	PTR DATA 0/	
72	PTP SYS RDY/	PTP System Ready
73	PTR SIG GND 4	Ground
74	TAPE CHAD ERR/	PTP Status
75	PTR SIG GND 3	Ground
76	UNUSED GND	Ground
77	PTR SIG GND 2	Ground
78	PTP RDY/	PTP Ready
79	PTR SIG GND 1	Ground
80	PTR SYS RDY/	PTR System Ready
81	PTP SIG GND 3	Ground
82	PTR RDY	PTR Ready
83	PTP SIG GND 1	Ground
84	PTP SIG GND 2	Ground
85	PROM DATA 0/	PROM Data Input
86	PROM DATA 1/	
87	PROM DATA 2/	PROM Data Input
88	PROM DATA 3/	
89	PROM RD DAT/	PROM Read Data
90	PROM RD STAT/	PROM Read Status
91	PROM DATA 5/	
92	PROM DATA 6/	PROM Input Data
93	PROM DATA 7/	
94	PROM DATA 4/	
95	LPT DATA STRB/	LPT Data Strobe
96	PROM WRT DAT PLS/	PROM Write Data Strobe
97	LPT GND	Ground
98	PROM CTL PLS/	PROM High Address-Control Strobe
99	UNUSED GND	Ground
100	PROM ADR PLS/	PROM Low Address Strobe

5.3.3 SUMMARY OF DEDICATED I/O ADDRESSES

Table 5-9 summarizes those I/O addresses which are dedicated to specific functions within the INTELLEC MDS System.

5.4 OPERATING CHARACTERISTICS: MONITOR MODULE

The AC and DC characteristics of all major signals that appear at the edge connectors will be listed in this section. This information, however, will be supplied by Intel.

5.4.1 AC CHARACTERISTICS

See Tables 5-10a and 5-10b.

5.4.2 DC CHARACTERISTICS

See Tables 5-11 and 5-12. Power requirements are cited below:

	TYP	MAX
V_{CC} +5 VDC ±5%	2.13A	2.97A

Table 5-9

SUMMARY OF DEDICATED I/O ADDRESSES

ADDRESS	INPUT	OUTPUT
ØØFF	*Real Time Clk	*Enable RTC
ØØFE	Reserved	*Override
ØØFD	Reserved	*Store Cur Level
ØØFC	*Int Mask	*Int Mask
ØØFB	LPT Status	LPT Control
ØØFA	INT Status	LPT Data
ØØF9	PT Status	PT Control
ØØF8	PTR Data	PTP Data
ØØF7	CRT Status	CRT Control
ØØF6	CRT Data	CRT Data
ØØF5	TTY Status	TTY Control
ØØF4	TTY Data	TTY Data
ØØF3	————	Monitor Int cntrl
ØØF2	————	PROM prog address LSB
ØØF1	PROM prog status	PROM prog high addr/ control
ØØF0	PROM prog data	PROM prog DATA

*Implemented on Front Panel Control Module in INTELLEC MDS system.

Table 5-10a

MONITOR MODULE MDS BUS AC CHARACTERISTICS

PARAMETER	OVERALL MIN	OVERALL MAX	DESCRIPTION INPUT REQUIREMENTS	REMARKS
t_{AS}	50		Address set-up time to command	
t_{AH}	50		Address hold time from command	
t_{DS}	50		Data set-up time to command, WRT	
t_{DHW}	50		Data hold time from command, WRT	
t_{SEP}	141		Command separation	
t_{WC}	t_{ACC}		Command width	
t_{XKCO}	0		Command turn off delay from XACK/	
t_{BCY}	100		Bus Clock cycle time	
t_{BW}	25		Bus Clock low and high periods	
t_{CC}	101.725		Com. clock cycle time ±0.05%	
t_{CW}	25		Com. clock low and high periods	
			OUTPUT LIMITS	
t_{DH1}		76	INH1/ Delay from ADRx/, INH2/	
t_{XKO}		141	XACK turn off delay	
t_{DHR}	0		Data hold from read commend	
t_{ACC}		3424	XACK/ delay from VAL RD DATA	Pulsed output to PROM prgrm or LPT
		90+ t_{ACCPP}		t_{ACCPA}=access by PROM prog.
		1811		Other commands
t_{XKD}	50		XACK/ delay from VAL RD DATA	

Table 5-10b

MONITOR MODULE EXTERNAL INTERFACE AC CHARACTERISTICS

PARAMETER	OVERALL MIN	OVERALL MAX	DESCRIPTION OUTPUT LIMITS	REMARKS
t_{EWD}		95	Delay from IOWC/	
t_{EFSU}	1200		PTP fro set-up to PTPADV↑	
t_{EDRST}		62	PTR DRV reset from PTR RDY/↑	
t_{EDSD}	1236		OUTDATx/ set up to strobe	
t_{ESTB}	746		Strobe width	
t_{EDMO}	697.8		OUTDATx/ hold from strobe	
t_{DII}	0		IORC/ to PROM RD	
t_{AK-AK}		50	PROM ACK Delay	
			INPUT LIMITS	
t_{EDRDY}	1811		PTR RDY/ delay from PTR DRV	
t_{EPRRD}		140	PTR RDY/ off from PTR BRV/↑	
t_{EDSR}		100	Valid Data from IORC/	
t_{EDHI}	100		PROM Data hold	
t_{EDSI}	0		PROM Data set up	
t_{EDHR}	0		Hold from IORC/	
t_{EDVR}	0		Valid after IORC/ lead	

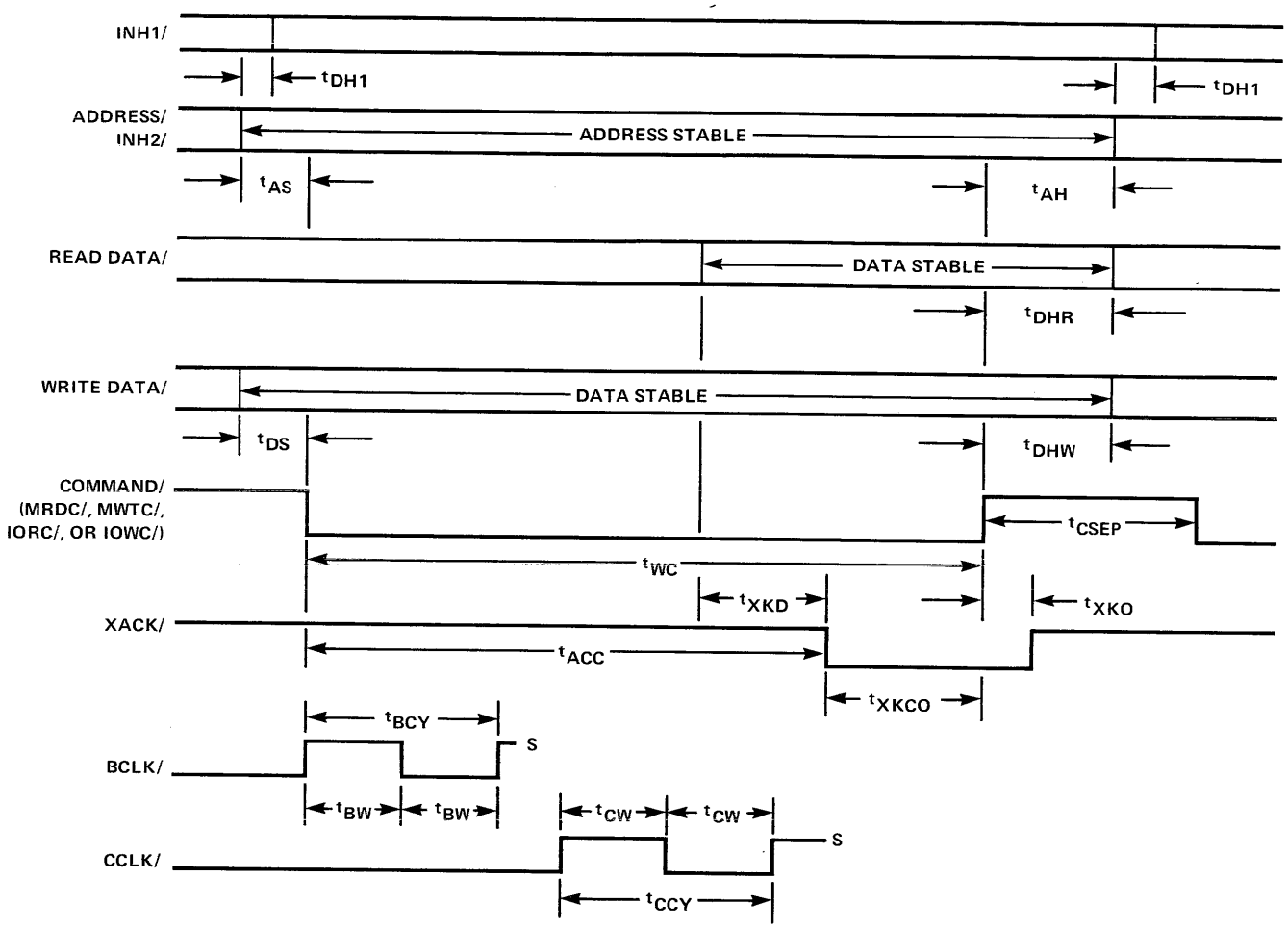


Figure 5-20. Command Timing

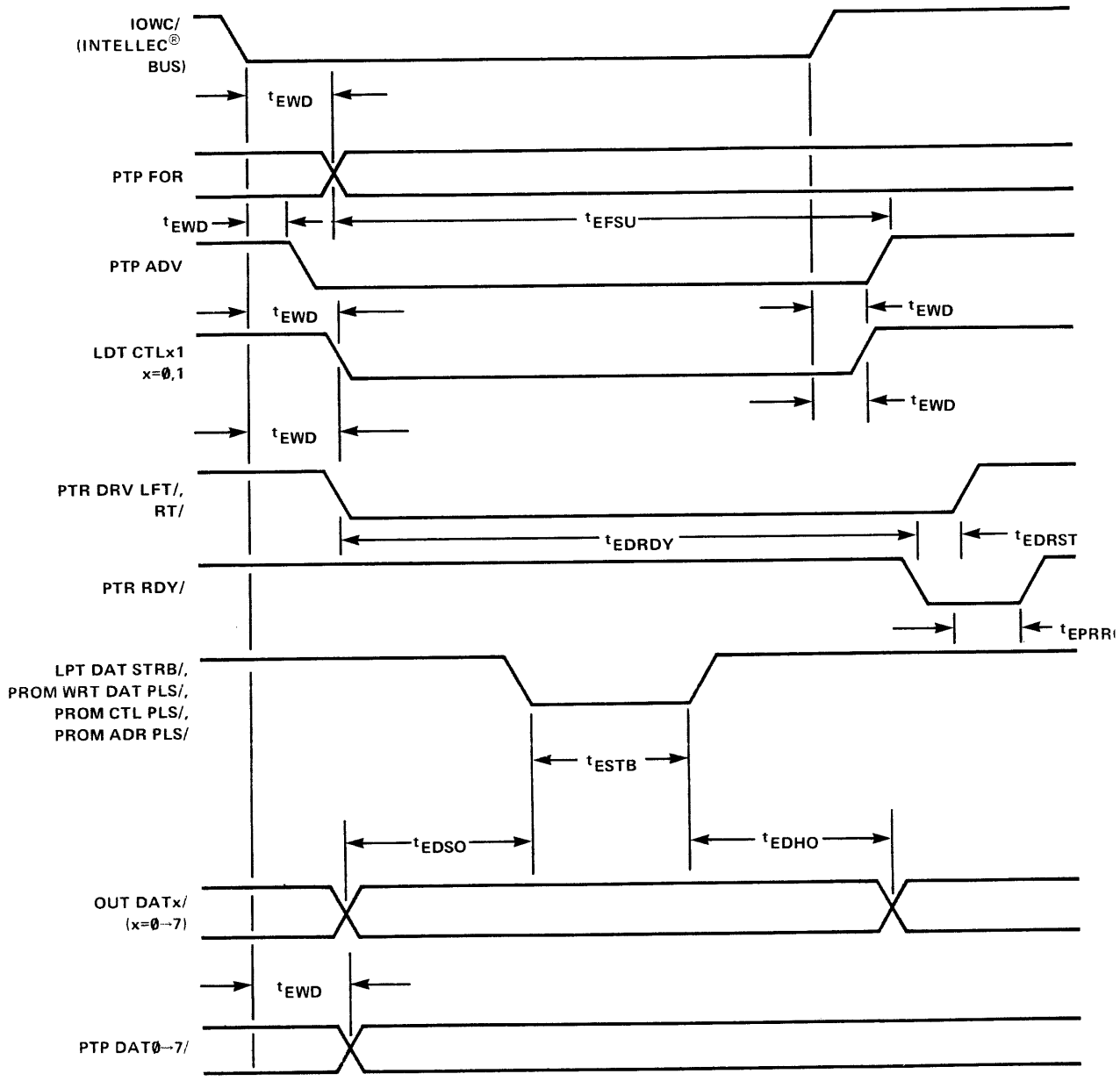


Figure 5-21. External I/O Write Timing

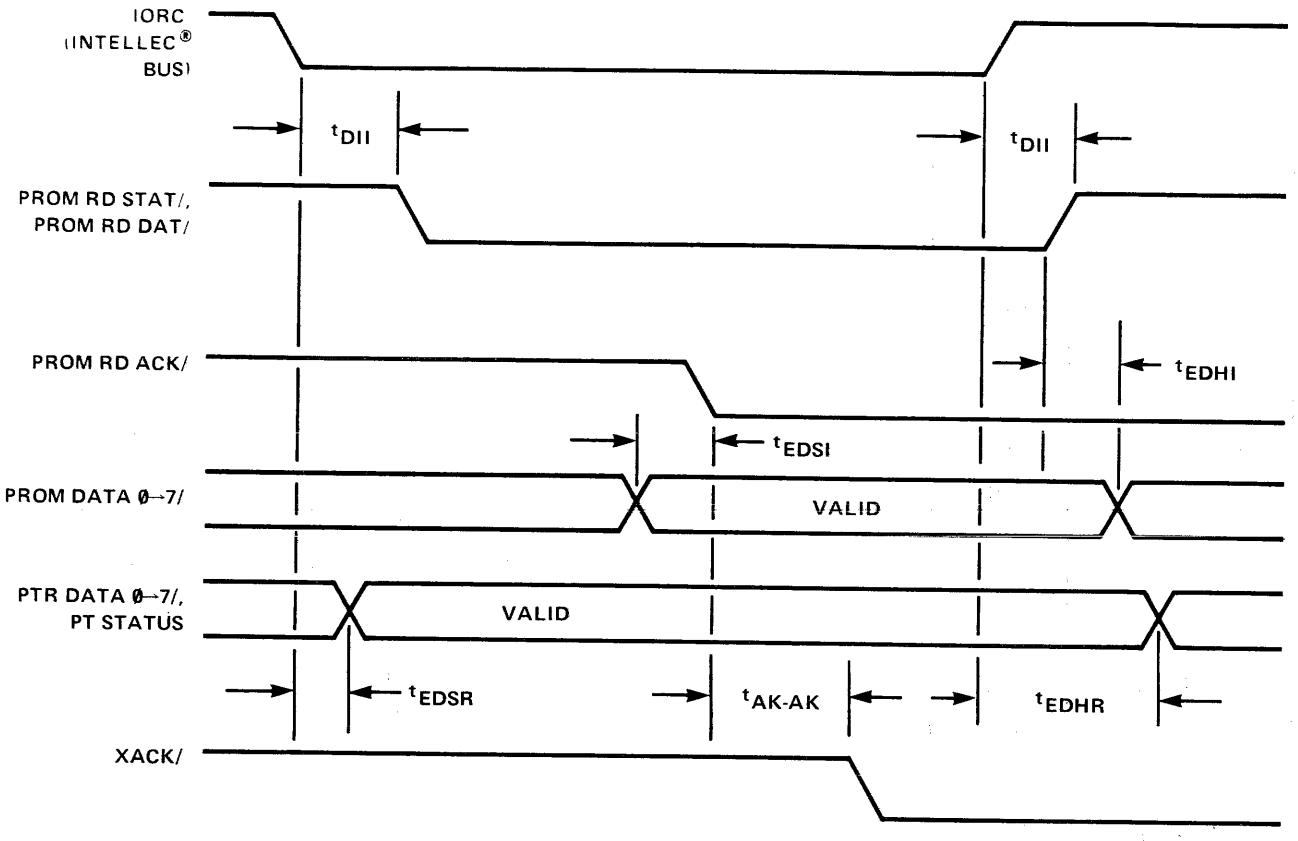


Figure 5-22. External I/O Read Timing

Table 5-11

MONITOR MODULE DC CHARACTERISTICS (INTELLEC® Bus)

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	PARAMETER			
				MIN	TYP	MAX	UNITS
ADRØ/→ADRF/, INH2/	V_{IL}	Input Low Voltage				0.65	V
	V_{IH}	Input High Voltage		2.2			V
	I_{IL}	Input Current at V_{IL}	V_{IL} = 0.45V			-0.52	mA
	I_{IH}	Input Current at V_{IH}	V_{IH} = 2.4V			40	μF
	C_L	Capacitive Load				30	pF
IOWC/, IORC/	V_{IL}					0.85	V
	V_{IH}			2.2			V
	I_{IL}		V_{IL} = 0.45V			-0.5	mA
	I_{IH}		V_{IH} = 5.25V			20	μF
	C_L					10	pF
XACK/	V_{OL}		I_{OL} = 16 mA	2.4		0.4	V
	V_{OH}		I_{OH} = -5.2 mA				V
	I_{LH}		High Z V_0 = 2.4V			40	μA
	I_{LL}		High Z V_0 = 0.4V			-40	μA
	C_L					15	pF
INT3/, INH1	V_{OL}		I_{OL} = 16mA			0.4	V
	I_{OH}		V_{OH} = 12V			50	μA
	C_L					15	pF
DATØ/→DAT7/	V_{OL}		I_{OL} = 32mA			0.5	V
	V_{OH}		I_{OH} = -5.2mA	2.4			V
	V_{IL}					0.8	V
	V_{IH}			2.2			V
	I_{IL}		V_{IL} = 0.45V			280	μA
	I_{IH}		V_{IH} = 2.4V			540	μA
	C_L					15	pF
DAT8/→DATF/	V_{OL}		I_{OL} = 32mA			0.4	V
	V_{OH}		I_{OH} = -5.2mA	2.4			V
	I_{IL}		High Z V_0 = 0.4			-40	μA
	I_{IH}		High Z V_0 = 2.4			40	μA
	C_L					15	pF
CCLK/, INIT/	V_{IL}					0.85	V
	V_{IH}			2.0			V
	I_{IL}		V_{IL} = 0.45V			-0.25	mA
	I_{IH}		V_{IH} = 5.25V			10	μA
	C_L					15	pF

Table 5-12

MONITOR MODULE DC CHARACTERISTICS (EXTERNAL INTERFACE)

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	PARAMETER			
				MIN	TYP	MAX	UNITS
PTP DATØ→7/,	V_{OL}	Output Low Voltage	I_{OL} = 32mA			0.4	V
OUT DATØ→7/,	V_{OH}	Output High Voltage	I_{OH} = -5.2mA	2.4			V
PROM RD STAT/,	C_L	Capacitive Load				15	pF
PROM PD DAT/, PTP ADV, PTP FOR							
INITIALIZE/	V_{OL}		I_{OL} = 16mA			0.4	V
	V_{OH}		I_{OH} = -5.2mA	2.4			V
	C_L					15	pF
PROM WRT DAT PLS/,	V_{OL}		I_{OL} = 48mA			0.4	V
PROM CTL PLS,	V_{OH}		I_{OH} = -1.2mA	2.4			V
PROM ADR PLS,	C_L					15	pF
LPT DATSTRB, PTR DRV RT, PTR DRV LFT, LPT DTLØ, LPT CTL1							
CRT Tx DATA/	V_{OL}		Jumper POS ⑦ I_{OL} = 10mA			-5	V
	V_{OH}		Jumper POS ⑦ I_{OH} = 50mA	5			V
	V_{OL}		Jumper POS ⑥ I_{OL} = 16mA			0.4	V
	V_{OH}		Jumper POS ⑥ I_{OH} = -5.2mA	2.4			V
	C_L					15	pF
TTY Tx Data,	I_{OL}	Output Low Current	V_{OL} = -10V, output is off			20	μA
TTY RDR CTL	V_{OH}	Output High Voltage	I_{OH} = -40mA	2.7			V
	C_L					15	pF
PT Status signals,	V_{IL}	Input Low Voltage				0.8	V
LPT ACK/,	V_{IH}	Input High Voltage		2			V
LPT BUSY,	I_{IL}	Input Current at V_{IL}	V_{IL} = 0.4V			-6.4	mA
LPT STAT1,	I_{IH}	Input Current at V_{IH}	V_{IH} = 2.4V			160	μA
PROM RD ACK/	C_L					15	pF
PROM DATA Ø→7/,	V_{IL}					0.8	V
PTR DATA Ø→7/,	V_{IH}			2			V
	I_{IL}		V_{IL} = 0.4V			-1.6	mA
	I_{IH}		V_{IH} = 2.4V			40	μA
	C_L					15	pF

Table 5-12

MONITOR MODULE DC CHARACTERISTICS (EXTERNAL INTERFACE) (continued)

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	PARAMETER			
				MIN	TYP	MAX	UNITS
CRT Rx DATA/	V_{IL}		Jumper POS ⑦			-3	V
	V_{IH}		Jumper POS ⑦	+3			V
	I_{IL}		Jumper POS ⑦ V_{IL} = -25V			-6	mA
	I_{IH}		Jumper POS ⑦ V_{IH} = 25V			9	mA
	V_{IL}		Jumper POS ⑥			0.6	V
	V_{IH}		Jumper POS ⑥	2			V
	I_{IL}		Jumper POS ⑥ V_{IL} = 0.45V			-6	mA
	I_{IH}		Jumper POS ⑥ V_{IH} = 2.4V			200	μA
	C_L					50	pF
TTY Tx DATA	V_{IL}					-5	V
	V_{IH}			3			V
	I_{IL}		V_{IL} = -12V			-20	mA
	I_{IH}		V_{IH} = +12V			65	mA
	C_L					0.15	μF
	V_{ILZ}	Input Voltage Open Circuit	I_{IL} = 0mA			-6	V

Chapter 6
THE 16K RAM MODULE

The RAM Module has been designed to provide the INTELLEC MDS System with 16,384 (16K) X 8-bit words of dynamic random access memory (read/write). Up to four RAM Modules can be used in the INTELLEC MDS System, providing the system with 65,536 words of read/write memory.

Although the RAM Module has been designed primarily to support the INTELLEC MDS System, its general-purpose architecture allows its inclusion in most 8-bit or 16-bit computer systems. Two 16K X 8-bit modules can be paired (i.e., both modules will respond to the same 16-bit memory address) to provide a 16K X 16-bit storage capacity. Byte selection on one module will send/receive its data byte (the low-order byte of the 16-bit data word) to/from one set of lines on the system data bus (DATØ–7) while the byte selection logic on the other module in the pair will send/receive its byte (the high-order byte) to/from another set of lines on the data bus (DAT8–F).

The RAM Module can complete a read cycle in 735 ns (worst case), and a write cycle in 1.36 μs (worst case). In addition, all of the logic required to refresh the dynamic RAM elements (at 24-μs intervals) is included on the module.

The module is implemented on a single 12-in. X 6.75-in. printed circuit board. The module requires only DC voltages at levels of +5, −10, and +12 VDC.

6.1 FUNCTIONAL ORGANIZATION OF THE 16K RAM MODULE

In order to describe its operation, the RAM Module has been divided into five functional units:

(1) Memory storage block
(2) Module select block
(3) Address control block
(4) Operation control block
(5) Read/write buffers

A block diagram of the RAM Module, illustrating the five functional units and their interrelationships, is provided in Figure 6-1.

The *memory storage block* consists of thirty-two 2107 dynamic random-access-memory (RAM) elements. Each 2107 element stores 4096 bits of data. The 32 elements are organized into four banks of eight elements each, providing a total storage capacity of 16,384 X 8-bit words per RAM Module. Because the RAM elements are dynamic, they require periodic refreshing, which is provided by logic on the module (see operational control block, below).

The *module select block* decodes the two most significant address bits (ADRE/ and ADRF/) to determine whether the module has been selected. The module select block informs the operation control block when it is selected. The two address bits can uniquely select any one of four RAM modules (or module-pairs when 16-bit data storage is employed). The particular 2-bit code that a module will respond to is determined by jumper connections on the card.

The remaining 14 bits of the address are applied to the *memory control block*. Address bits 0–5 (ADRØ/–ADR5) identify one of 64 rows to be accessed within each 4K X 1-bit memory element, while address bits 6–11 (ADR6/–ADRB/) identify one of 64 columns to be accessed. Thus, these 12 address bits uniquely identify one bit location within each of the 2107 RAM elements. The remaining two address bits, 12 and 13 (ADRC/ and ADRD/), enable one of four banks on the selected RAM module(s). Each bank consists of eight 2107B RAM elements, with each element within the enabled bank contributing one addressed bit location to the 8-bit data byte that is read from or written into the selected RAM module.

The *operation control block* accepts memory read (MRDC/) and write (MWTC/) commands and executes the command as soon as the module is not

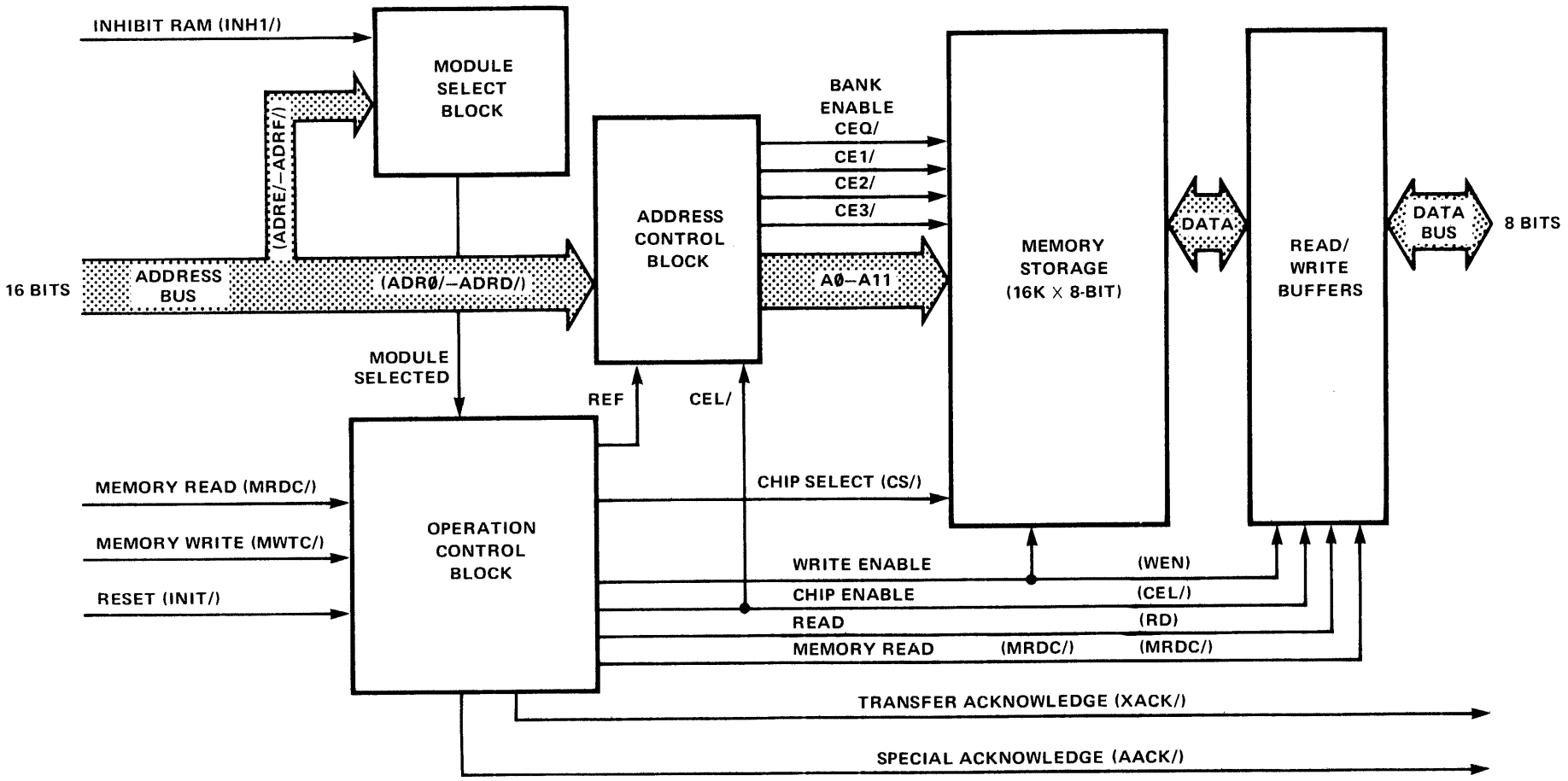


Figure 6-1. RAM Module Function Block Diagram

busy; that is, after the current refresh cycle, if one is in progress. The operation control block generates acknowledge signals for the CPU (AACK/ and XACK/), and maintains its read/write command (RD), BUSY/, chip enable (CEL/), chip select (CS/), and write enable (WEN) latches. The operation control block is also responsible for periodically refreshing the dynamic RAM elements. Approximately once every 12 μs, the operation control block refreshes (REF goes true) one of 64 rows (64 bits) on each of the 32 RAM elements. A refresh cycle, however, will be postponed until after a read or write cycle, if one is in progress at the end of the current 12-μs interval.

The *read/write buffers*, as their name implies, buffer data that is being written into or read from the RAM Module. When two RAM Modules are paired (i.e., both have the same 2-bit module select code) to implement storage of 16-bit data words, byte selection logic is added to this section. The byte selection logic directs/accepts a data byte to/from eight different pins on the edge connector. Thus, the low-order byte of the 16-bit data word is read from/written into one module of the pair, while the high-order byte is read from/written into the other module.

6.2 RAM MODULE: THEORY OF OPERATION

In this section we provide a detailed theory of operation description for the RAM Module. We begin by describing the physical memory implementation (Section 6.2.1) and address decoding (Section 6.2.2). We will then step through memory write (Section 6.2.3), read (Section 6.2.4), and refresh (Section 6.2.5) cycles to show how the address control, operation control, and read/write buffer blocks function together to execute the specified operation.

The schematic (6 sheets) for the RAM Module is provided in Figure 6-5, located in Section 6.2.6.

6.2.1 PHYSICAL MEMORY IMPLEMENTATION

The actual memory on the RAM Module consists of 32 Intel 2107 dynamic random-access-memory (RAM) elements. Each 2107 element has a 4096-bit capacity. The 32 RAM elements are organized into four banks of eight elements each. The eight RAM elements each contribute one bit to an addressable location. Thus, the RAM Module has a total storage capacity of 16,384 X 8-bit words (four banks times 4096 X 8 bits per bank). The four memory banks are shown on sheets 3, 4, 5, and 6 of the module schematic, Figure 6-5.

Each 2107 RAM element has 12 address inputs (AØ–A11), a chip enable input (CE), inverted write enable (WE) and chip select (CS) inputs, a data input (DI) and a data output (DO), as well as power inputs of +12 VDC, +5 VDC, –5 VDC, and ground. Address bits AØ–A5 identify one of 64 rows, while bits A6–A11 identify one of 64 columns; the intersection of row and column addresses defines the bit to be accessed. WE indicates whether data is being read or written; CE provides a timing reference when this memory bank has been selected; and CS differentiates between read/write cycles and refresh cycles (i.e., CS enables RAM outputs during read cycles and RAM inputs during write cycles).

As previously mentioned, two modules can be paired to provide a 16K X 16-bit storage capacity. Both modules must have the same module select code (specified by address bits ADRE/ and ADRF/), but must have complementary byte selection logic; that is, one module will store the high-order byte of each 16-bit word, while the other module stores the low-order byte of the data word.

By combining more than one module (or module pair) in a system, memory size can be expanded up to 65,536 (64K) 8- or 16-bit words.

6.2.2 MEMORY ADDRESS DECODING

The two most significant address bits (ADRE/ and ADRF/) select the proper module (or module pair). The remaining 14 address bits (ADRØ/–ADRD/) specify the memory bank, as well as the column and row within the memory elements.

The module select logic consists of a 3205 three-to-eight decoder, three 74S00 NAND gates, and a 7404 inverting buffer, as shown on sheet 1 of the module schematic, Figure 6-5. The address control logic consists of two 74L93 counters, one 6-bit

3404 buffer, a 3205 decoder, and assorted gating circuits, as shown on sheet 2 of the module schematic.

The two most significant address bits (ADRE/ and ADRF/) and the inhibit RAM signal (INH1/), are applied to the three address inputs of the 3205 decoder. The decoder's three enable inputs (E1, E2, and E3) are permanently active. Only the four high-order decoder outputs are used; consequently, the module cannot be selected unless INH1/ is high (inactive). Only one of the four decoder output lines is closed. The closed decoder output line determines which code on lines ADRE/ and ADRF/ will select the module (see Table 6-1). The decoder output is inverted and applied to one input of 74S00 NAND gate. The other input is supplied by the OR of memory read (MRDC/) and memory write (MWTC/). The output of this NAND gate is then inverted by another 74S00 section. The module is selected only when the level on the output line from this final 74S00 section is high. The high output is fed to the operation control logic (also shown on sheet 1 of the module schematic).

The remaining 14 address bits are fed to the address control logic. The six least significant bits (ADRØ/–ADR5/) are applied to one of the negative inputs of 8210 AND gates. The other input is the active-high refresh signal (REF). Consequently, the six low-order address bits are gated through only when a refresh cycle is *not* in progress (see Section 6.2.5). The address bits are NORed with the outputs of the two 74L93 counters and made available to the 2107 RAM elements. These six address bits specify the row to be accessed within each RAM.

During refresh cycles, the address bits are not gated through. Instead, the output of the 74L93 counters supplies the row address to the RAM elements. The two counters, acting as a single 8-bit binary counter, are incremented at the end of each refresh cycle (i.e., by the trailing edge of REF).

The next six address bits (ADR6/–ADRB/) specify the column to be accessed within each RAM element. During read or write cycles, these address bits are inverted and buffered by the 3404 circuits, and made available to the most significant address inputs of the RAM chips (A6–A11). During refresh cycles, however, the high level on the REF

Table 6-1

MODULE SELECT ADDRESS BITS

JUMPER CONNECTIONS	DECODER OUTPUT (PIN)		*SELECTION CODE	
			ADRE/	ADRF/
7-8	7	(7)	1	1
7-9	6	(9)	0	1
7-6	5	(10)	1	0
7-5	4	(11)	0	0

*NOTE: The levels on these address lines are active-low; that is, l=logical 0 and 0=logical l.

line (applied to the negative write enable inputs of the buffer) prevents the output of the 3404 buffers from changing. These address bits are not used by the RAM elements during refresh cycles, because the entire row specified by the six least significant bits (AØ–A5) is refreshed. The 2107 RAM specifications, however, dictate that address lines A6–A11 not be permitted to change during the refresh operation; thus, the reason for disabling the 3404 buffers.

Address bits 12 and 13 (ADRC/ and ADRD/) are buffered in 3404 devices and used to select one of four memory banks. Each bank includes eight 2107 RAMs. Only one bank is selected during any read or write cycle. Each RAM element in the selected bank contributes one bit position to the byte being accessed. ADRC/ and ADRD/ feed the two least significant inputs to the 3205 decoder (AØ and A1). The other decoder address input, A2, is tied to ground. Thus, only the four least significant inverted decoder outputs are used. Each of these four outputs is NORed with the active-low refresh signal (REF/). If the chip enable latch is set (i.e., if CEL/ is true), the appropriate chip enable signal (CEØ, CE2 or CE3) goes true. CEØ, CE1, CE2, and CE3 each enable one of the four memory banks (as shown on sheets 3–6 of the module schematic). During refresh cycles, the REF/ signal enables all four memory banks.

To summarize: During read or write cycles, one memory bank (eight RAMs) is enabled by address bits ADRC/ and ADRD/. Address lines ADRØ–ADR5/ specify the row within the RAM elements

of the selected bank, and address lines ADR6/–ADRB/ specify the column. The intersection of row and column addresses defines the bit position to be accessed in each of the eight selected RAM elements. During refresh cycles, all four banks are enabled, and the column address bits are prevented from changing. The row address is incremented by one at the end of each refresh cycle. Consequently, each refresh cycle rejuvenates one complete row (i.e., 64 bits) within each RAM element on the module. Sixty-four refresh cycles are required to refresh the entire module.

6.2.3 MEMORY WRITE CYCLE

A memory write operation is initiated by the Central Processor. The CPU places a 16-bit memory address on the address bus, places (within 500 ns) an 8-bit data byte on the data bus (or 16-bit if paired-modules are used with a 16-bit processor), and issues the memory write command (MWTC/). The two most significant address bits (ADRE/ and ADRF/) select the proper module (or module-pair), while the remaining 14 address bits (ADRØ/–ADRD/) are available to the address control logic, as described in Section 6.2.2. MWTC/ is applied to the operation control logic, shown on sheet 1 of the module schematic. The data byte is available at the read/write buffers, shown on sheet 2 of the module schematic.

MWTC/ is fed to the D-input of the write command latch in the operation control block. The active-low selected signal (A9-6) from the module select block (see Section 6.2.2) enables two 8093 drivers that will subsequently enable active-low levels on the acknowledge lines, AACK/ and XACK/. If a refresh cycle is not in progress, and if the last high-to-low transition of the refresh request latch output (A14-9) did not occur within the last 600 ns (see Section 6.2.5), the AACK/ special acknowledge signal is asserted immediately. Otherwise, it will go active when a 9602 one-shot (600 ns pulse at A13-9) times out. XACK/ can be asserted later in the cycle, as described in the latter portion of this section.

The selected signal from the module select block is inverted (i.e., an active-high level appears at A9-8) and is applied to the clock input of the command latch (pin A3-3 of a 74S74 D-type flip-flop), causing the latch to set. The $\overline{Q}$ output is ORed with the $\overline{Q}$ output of the refresh request latch. The active-high result is fed to one input of a 74S00 NAND gate. The other input is supplied by the busy latch (BUSY/). If the module is busy (i.e., if a refresh cycle is in progress), the write cycle will wait until the refresh is completed. That is, the write cycle will wait until the BUSY/ line goes high, enabling the 74S00 gate (A1-6). The enabled 74S00 output is gated through to the clock input of the command/$\overline{\text{refresh}}$ latch (pin A3-11 of another 74S74 flip-flop) and to the input line of a PE 9825 delay line circuit (DL1; 10-ns taps). Assuming that a refresh request is not active, the D-input to the command/$\overline{\text{refresh}}$ latch will be high, causing the latch to set. The Q output of the latch is inverted (at A10-1) and asserted on the chip select (CS/) line. CS/ is applied to the active-low chip select input on each of the RAM elements. CS/ specifies that a read or write cycle is in progress; not a refresh cycle.

Forty nanoseconds after the command/$\overline{\text{refresh}}$ latch sets, a high level appears on the 4th tap of the delay line circuit (D1). This 40-ns delay allows sufficient time for the simultaneous occurrence of a command and a refresh request to be resolved by the command/$\overline{\text{refresh}}$ latch. The 4th delay circuit tap is NANDed with the Q output of the command/$\overline{\text{refresh}}$ latch. The resultant active-low signal (twice inverted) pre-sets the "previous cycle inhibit" latch (A6-4), pre-resets the chip enable (CEL/) latch (A7-1), clocks the write command latch, and sets the pulse delay line (PDL) latch (a 74S40 and a 74S10 section wired together as an S-R latch).

The write command latch is clocked to the reset state (i.e., RD goes false) because the memory write command (MWTC/) is active-low and it feeds the D-input of the latch. This latch will remain reset until it is clocked set by a memory read cycle or until it is pre-set by a refresh cycle (REF/).

CEL/ goes true when the chip enable latch is reset. CEL/ activates one of the chip enable signals (CEØ, CE1, CE2, or CE3) that specifies the selected memory bank, as described in Section 6.2.2.

The output of the Pulse Delay Line (PDL) latch is applied to the input line of a PE 9828 delay line circuit (DL3; 20-ns taps). The positive-going edge of the PDL latch output will appear at the output

line of the PE 9828 circuit 200 ns later. The output is then inverted and applied to the input line of a second PE 9828 delay line circuit (DL4). This negative-going edge will appear at the output line of the second PE 9828 circuit 200 ns later (i.e., 400 ns after the PDL latch was set). The inverted level is inverted again (it is now a positive-going edge) and applied to the input line of a third PE 9828 delay line circuit (DL2). The positive-going edge will appear at the DL2 output line 200 ns later (600 ns after the PDL latch is set). The PDL latch will be reset 520 ns into this 600-ns sequence, causing a negative-going edge to start down through the three delay line circuits. These positive and negative-going transitions will appear at the various taps on the three delay line circuits after appropriate delays (20 ns between each two taps). The transitions that appear at these taps provide sequencing control for the write, read and refresh operations performed by the RAM Module. In the following paragraphs, we will refer all events to the setting or resetting of the PDL latch.

As the PDL latch is set, the BUSY/ latch is preset. The Q output (BUSY/) disables the 74S00 gate (A1-5) that allows a command or refresh request to initiate a new cycle within the operation control block, as previously described.

Four-hundred and eighty (480) nanoseconds after the PDL latch was set, the positive-going transition that appears at the fourth tap of DL2 clocks the write enable (WEN) latch. Because the D-input of the WEN latch is fed by the high $\overline{Q}$ output of the write command latch, the WEN latch is clocked to the set state. The Q output of the write enable latch (WEN) latches the data to be written into the 3404 write buffers, as shown on sheet 2 of the module schematic. The buffer outputs are available to the data inputs of the eight selected RAM elements. The $\overline{Q}$ output of the write enable latch (WE/) is applied to the active-low write enable input (WE) on each of the RAM elements, as shown on schematic sheets 3–6.

The output from the fourth tap of DL2 also clocks the transfer acknowledge flip-flop (A8-3). Because a refresh cycle is not in progress, the D-input to this flip-flop (REF/) will be high; consequently, the flip-flop will be clocked set. The low $\overline{Q}$ output feeds the XACK/ line. XACK/, however, will only be asserted on the system bus (P1-23) if jumper pad 1-2 is connected. In the INTELLEC MDS System, XACK/ is not normally required because the advance acknowledge signal, AACK/, has already acknowledged the RAM access as previously described. XACK/ is used for bus master modules other than the 8080-based CPU Module (e.g., a CPU Module that does not include an 8080 processor, or a DMA Module). The high Q output from the transfer acknowledge flip-flop clocks a 74LS74 flip-flop (A8-11) to the reset state. The Q output of this 74LS74, in turn, disables the special acknowledge (AACK/) signal. The low Q output from the transfer acknowledge flip-flop pre-resets the command latch (A3-1) in anticipation of the next command or refresh cycle.

The MACK/ signal (pin P1-27) can be jumper connected to the XACK/ line to acknowledge 16-bit transfers, when two RAM Modules are paired together for 16-bit storage configurations (see Section 6.3.1).

Five-hundred and twenty (520) nanoseconds after the PDL latch was set, the PDL latch is reset by the output of a 7400 NAND gate (A11-8). A negative-going edge starts down through the three delay line circuits. Forty (40) nanoseconds later (560 ns after the PDL latch was set, the 7400 gate is deactivated. The PDL latch, however, will already have been reset and will remain so until the next cycle.

Five-hundred and sixty (560) nanoseconds after the PDL latch was set (40 ns after it was reset), the "previous cycle inhibit" latch is clocked reset.

The negative-going edge that was input to DL3 when the PDL latch was reset, is inverted 200 ns later when it is fed into DL4. Consequently, a positive-going edge will appear at the various taps on DL4 as the pulse is sequenced through this delay circuit. When the low-to-high transition occurs at the ninth tap of DL4 (i.e., 380 ns after the PDL latch was reset or 900 ns after the PDL latch was set), the chip enable latch (CEL/) is clocked. Because the "previous cycle inhibit" latch has already been reset, the output from the 7400 NAND gate at A5-6 is high. This gate output supplies the D-input of the CEL/ latch; consequently, the CEL/ latch is clocked reset.

Twenty (20) nanoseconds later (i.e., 400 ns after the PDL latch was reset), a high level appears at the

output line of DL4. This high level enables a 7400 NAND gate which pre-resets the write enable latch (WEN). The other input to the 7400 NAND gate is supplied by the fourth tap on DL2, which will exhibit a low level 80 ns later; thus disabling the gate after it has reset WEN.

Six-hundred (600) nanoseconds after the PDL latch was reset (or 1.2 μs after it was set), the high-to-low transition that finally reaches the output line of DL2 is inverted and NANDed with the high Q output from the write command latch. The resultant low signal is inverted and applied to the clock input of the busy latch. Because the D-input to the latch is high, it sets, asserting a high (inactive) level on the Q output (BUSY/). This completes the write cycle; the RAM Module can now perform a refresh cycle (see Section 6.2.5) or can accept a new write or read (see Section 6.2.4) command.

Timing for the memory write operation is illustrated in Figure 6-2.

6.2.4 MEMORY READ CYCLE

A memory read operation is initiated by the Central Processor. The CPU places a 16-bit memory address on the address bus and issues the memory read command (MRDC/). The CPU expects to receive an acknowledge signal and a data word from the RAM Module.

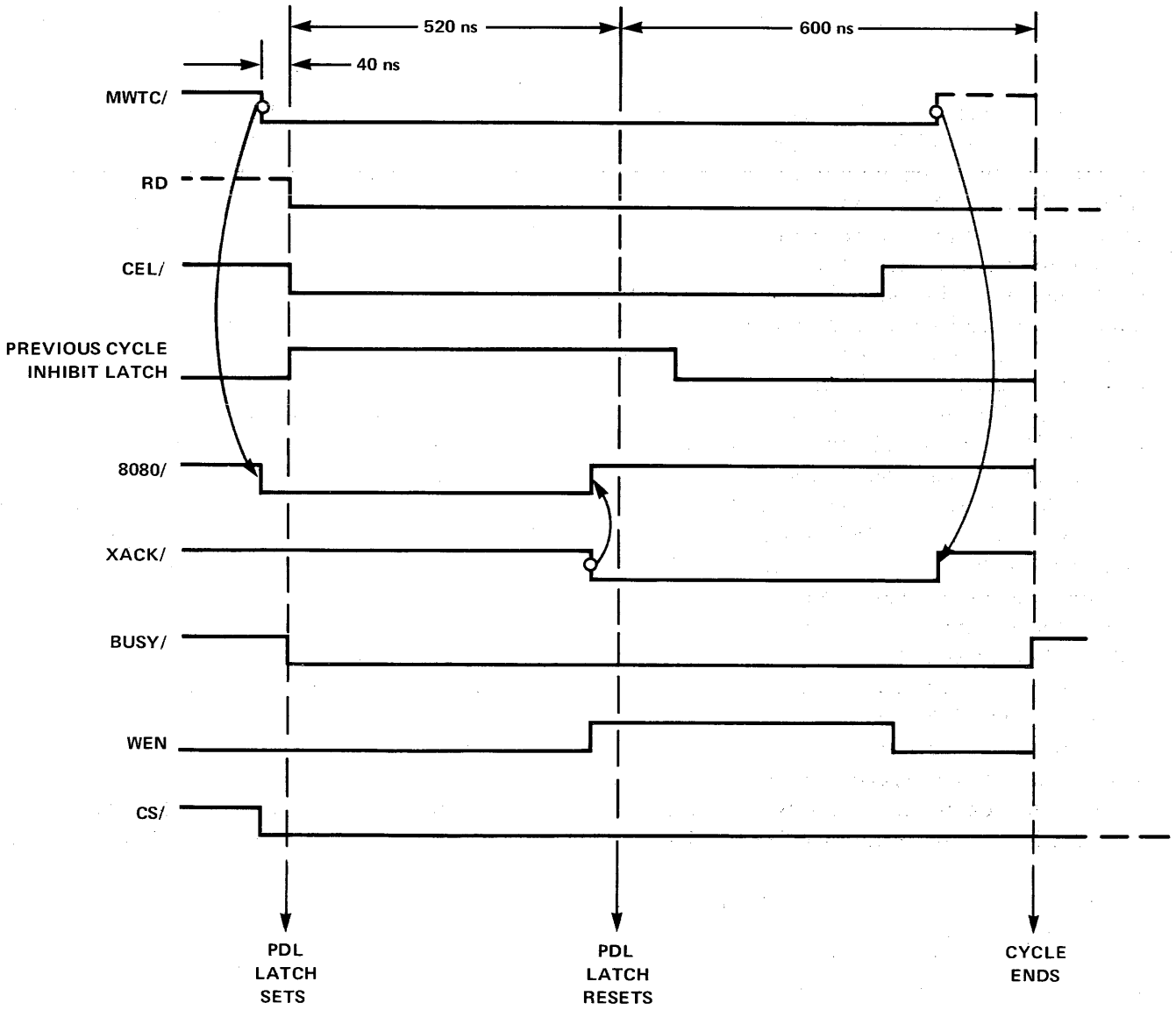


Figure 6-2. Memory Write Timing (RAM Module)

MRDC/ is fed to the module select logic (see Section 6.2.2) and to the DS1 input of the 3212 read buffer (sheet 2 of the module schematic).

Refer to sheet 1 of the module schematic, Figure 6-5. The active-low selected signal (A9-6) from the module select block enables two 8093 drivers that will subsequently enable active-low levels on the acknowledge lines, AACK/ and XACK/. If a refresh cycle is not in progress and if the last high-to-low transition of the refresh request latch output (A14-9) did not occur within the last 600 ns (see Section 6.2.5), the AACK/ special acknowledge signal is asserted immediately. Otherwise, it will go active when a 9602 one-shot (600-ns pulse at A13-9) times out. XACK/ can be asserted later in the cycle, as described in the latter portion of this section.

The selected cycle from the module select block is inverted (an active-high level appears at A9-8) and is applied to the clock input of the command latch (pin A3-3 of a 74S74 flip-flop), causing the latch to set. The $\overline{Q}$ output is ORed with the $\overline{Q}$ output of the refresh request latch. The active-high result is fed to one input of a 74S00 NAND gate. The other input is supplied by the busy latch (BUSY/). If the module is busy (i.e., if a refresh cycle is in progress), the read cycle will wait until the refresh is completed. That is, the read cycle will wait until the BUSY/ line goes high, enabling the 74S00 gate (A1-6). The enabled 74S00 output is gated through to the clock input of the command/$\overline{\text{refresh}}$ latch (pin A3-11 of another 74S74 flip-flop) and to the input line of PE 9825 delay line circuits (DL1; 10-ns taps). Assuming that a refresh request is not active, the D-input to the command/$\overline{\text{refresh}}$ latch will be high, causing the latch to set. The Q output of the latch is inverted (at A10-1) and asserted on the chip select (CS/) line. CS/ is applied to the active-low chip select signal input on each of the RAM elements. CS/ specifies that a read or write cycle is in progress; not a refresh cycle. CS/ disables the RAM outputs during refresh cycles.

Forty (40) nanoseconds after the command/$\overline{\text{refresh}}$ latch sets, a high level appears on the 4th tap of the delay line circuit (DL1). This 40-ns delay allows sufficient time for the simultaneous occurrence of a command and a refresh request to be resolved by the command/$\overline{\text{refresh}}$ latch. The 4th delay circuit tap is NANDed with the Q output of the command/$\overline{\text{refresh}}$ latch. The resultant active-low signal (twice inverted) pre-sets the "previous cycle inhibit" latch (A6-4), pre-sets the chip enable (CEL/) latch, clocks the write command latch and sets the pulse delay line (PDL) latch (a 74S40 and a 74S10 section wired together as an S-R latch).

The write command latch is clocked to the set state (RD goes true) because the memory write command (MWTC/) is false (high) and it feeds the D-input of the latch. RD is applied to the device select (DS2) input of the 3212 read buffer, as shown on sheet 2 of the module schematic.

CEL/ goes true when the chip enable latch is reset. CEL/ activates one of the chip enable signals (CEØ, CE1, CE2 or CE3) that specifies the selected memory bank, as described in Section 6.2.2. In addition, CEL/ provides a strobe input to the 3212 read buffer, whenever the refresh signal (REF/) is false. The combination of CEL/, RD and MRDC/ allows the 3212 read buffer to input the data byte, read from the selected memory elements, to the CPU.

The output of the pulse delay line (PDL) latch is applied to the input line of a PE 9828 delay line circuit (DL3; 20-ns taps).

The positive-going edge of the PDL latch output will proceed through the three delay line circuits exactly as described for a write operation (Section 6.2.3). The events that occur as a result of the delayed transition, however, are different. The PDL latch will be reset 520 ns after it was set.

As the PDL latch is set, the BUSY/ latch is preset. The Q output (BUSY/) disables the 74S00 gate (A1-5) that allows a command or refresh request to initiate a new cycle within the operation control block, as previously described.

Three-hundred and twenty (320) nanoseconds after the PDL latch was set, the chip enable latch (CEL/) is clocked to the set state; CEL/ goes false (high).

The write enable (WEN) latch is clocked by the fourth tap on DL2, 480 ns after the PDL latch was set. Because the D-input to the WEN latch (RD/) is low, the latch remains reset and WEN does not go true as it did during a write cycle.

The output from the fourth tap on DL2 also clocks the transfer acknowledge flip-flop (A8-3). Because a refresh cycle is not in progress, the D-input of this flip-flop (REF/) will be high; consequently, the flip-flop will be clocked set. The low $\overline{Q}$ output feeds the XACK/ line. XACK/, however, will only be asserted on the system bus (P1-23) if jumper pad 1-2 is connected. In the INTELLEC MDS System, XACK/ is not normally required because the advance acknowledge signal, AACK/, has already acknowledged the RAM access, as previously described. XACK/ is used for bus master modules other than the 8080-based CPU Module (e.g., a CPU module that does not include an 8080 processor, or a DMA Module). The high Q output from the transfer acknowledge flip-flop clocks a 74LS74 flip-flop (A8-11) to the reset state. The Q output of this 74LS74, in turn, disables the advance acknowledge (AACK/) signal. The low $\overline{Q}$ output from the transfer acknowledge flip-flop pre-resets the command latch (A3-1) in anticipation of the next command or refresh cycle.

The MACK/ signal (pin P1-27) can be jumper connected to the XACK/ line to acknowledge 16-bit transfers, when two RAM Modules are paired together for 16-bit storage configurations (see Section 6.3.1).

Five-hundred and twenty (520) nanoseconds after the PDL latch was set, the PDL latch is reset by the output of a 7400 NAND gate (A11-8). Forty (40) nanoseconds later (560 ns after the PDL latch was set), the 7400 gate is deactivated. The PDL latch, however, will already have been reset and will remain so until the next cycle.

Five-hundred and sixty (560) nanoseconds after the PDL latch was set (40 ns after it was reset), the "previous cycle inhibit" latch is clocked reset.

Five-hundred and eighty (580) nanoseconds after the PDL latch was set, the busy latch is clocked to the set state; BUSY/ goes false (high). This completes the read cycle. A refresh or write cycle can now begin. Notice that during a read cycle, BUSY/ is only true for 540 ns, while during a write cycle, BUSY/ remains true for 1.08 μs.

Timing for the memory read operation is illustrated in Figure 6-3.

6.2.5 REFRESH CYCLE

Approximately once every 12 μs, logic on the RAM Module requests a refresh cycle to rejuvenate the dynamic RAM elements. The refresh cycle, like the memory read and write cycles, is supervised by the operation control logic, shown on sheet 1 of the module schematic, Figure 6-5. Each refresh cycle accesses 64 bit positions (one row) within each of the thirty-two 2107 RAM chips on the module.

A low-to-high transition from the $\overline{Q}$ output of a 9602 one-shot (A13-7) clocks the refresh request latch to the set state, and re-triggers itself, unless the system initialization signal (INIT/) is true. The one-shot will again exhibit a low level on its $\overline{Q}$ output. Approximately 12 μs later, another low-to-high transition will request another refresh cycle.

The low-level on the $\overline{Q}$ output of the refresh request latch (A14-8) is applied to the D-input of the command/$\overline{\text{refresh}}$ latch (A3-12) and to one input of a 74S00 negative-OR gate. If the RAM Module is not busy, the output of this 74S00 section is gated through to the input line of a PE 9825 delay line circuit (DL1) and to the clock input of the command/$\overline{\text{refresh}}$ latch, causing it to reset. If the module is currently busy (i.e., if a memory read or write cycle is in progress), the refresh cycle will wait until the current cycle is completed.

The low Q output from latch A8-9 activates a 74S10 negative-OR gate that prevents an AACK/ acknowledge from being asserted by the RAM Module. The low level on the Q output of the command/refresh latch prevents the chip select signal (CS/) from being activated and is applied to the D-input of the XACK/ latch. The high level on the CS/ line disables the RAM outputs during refresh cycles. The Q output of the command/refresh latch is NANDed with the Q output from the refresh request latch. The resultant low output (REF/) pre-sets the write command latch (RD goes true). This prevents XACK/ from being generated during the refresh cycle.

Both REF/ and its inverted counterpart, REF, are available to the address control block (see Section 6.2.2).

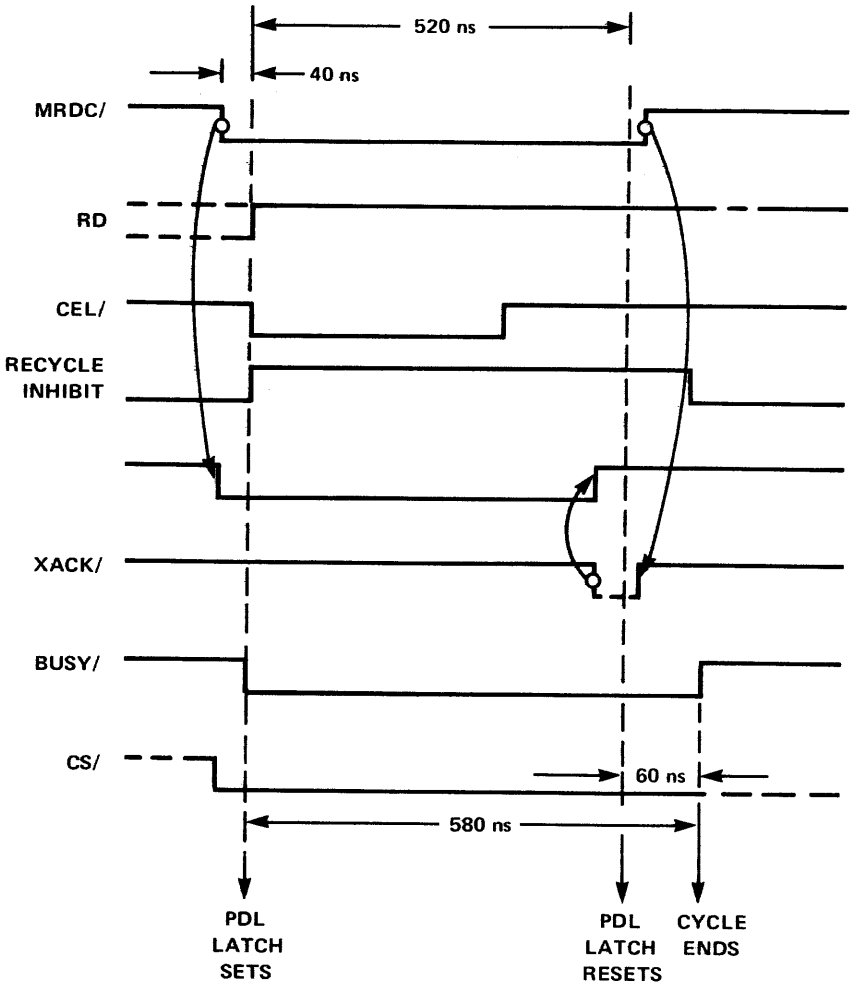


Figure 6-3. Memory Read Timing (RAM Module)

The $\overline{Q}$ output of the command/refresh latch is also NANDed (at A1-8,9,10) with the output line from delay line circuit, DL1. The DL1 output line goes true 100 ns after the command/$\overline{\text{refresh}}$ latch was clocked to the reset state. This 100-ns interval allows sufficient time for the two counters in the address control block to assert stable row address bits on the AØ–A5 inputs of the RAM elements, as described in Section 6.2.2.

The resultant low-level output at A1-8 (twice inverted) pre-sets the "previous cycle inhibit" latch (A6-4), pre-sets the chip enable (CEL/) latch and sets the pulse delay line (PDL) latch (a 74S40 and a 74S10 section wired together as an S-R latch).

CEL/ goes true when the chip enable latch is reset. CEL/ allows REF/ to activate all four memory banks (CEØ, CE1, CE2, and CE3 all go true), as described in Section 6.2.2.

The output of the pulse delay line (PDL) latch is applied to the input line of a PE 9828 delay line circuit (DL3; 20-ns taps). The positive-going edge of the PDL latch output will proceed through the three delay line circuits excatly as described for a write operation (Section 6.2.3). The events that occur as a result of the delayed transition, however, are different. The PDL latch will be reset, 520 ns after it was set.

As the PDL latch is set, the BUSY/ latch is preset. The Q output (BUSY/) disables the 74S00 gate (A1-5) that allows a command or refresh request to initiate a new cycle within the operation control block, as previously described.

Three-hundred and twenty (320) nanoseconds after the PDL latch was set, a high-to-low transition appears at the sixth tap on the second delay line circuit (DL4). This negative-going edge is inverted (at A10-6) and gated through to the clock input of the chip enable latch (CEL/) causing it to reset. CEL/ goes false (high).

The high level at A10-6 is also applied to one input of a 7400 NAND gate. The other input is supplied by the ninth tap on DL4. Because the ninth tap still exhibits a high level, the 7400 gate is activated. The resultant low output is ANDed (negative inputs) with the low level on the Q output of the command/$\overline{\text{refresh}}$ latch. The ANDed output is inverted, then used to pre-reset the refresh request latch (REF goes false). The high level that appears at the $\overline{Q}$ output is NANDed with the high $\overline{Q}$ output of the command/$\overline{\text{refresh}}$ latch. The resultant low signal triggers a 9602 one-shot (at A13-11). This one-shot will exhibit a low level on its $\overline{Q}$ output for 600 ns. This 600-ns signal prevents the special acknowledge signal (AACK/) from being prematurely generated during a subsequent memory read cycle.

Three-hundred and eighty (380) nanoseconds after the PDL latch was set, the pre-reset input to the refresh request latch is removed.

Five-hundred and twenty (520) nanoseconds after the PDL latch was set, the PDL latch is reset by the output of a 7400 NAND gate (A11-8). Forty (40) nanoseconds later, the 7400 gate is deactivated. The PDL latch, however, will already have been reset and will remain so until the next cycle.

Five-hundred and sixty (560) nanoseconds after the PDL latch was set (40 ns after it was reset), the "previous cycle inhibit" latch is clocked reset.

Five-hundred and eighty (580) nanoseconds after the PDL latch was set, the busy latch is clocked to the set state; BUSY/ goes false (high). This completes the refresh cycle.

Timing for the refresh operation is shown in Figure 6-4. Notice in the diagram how the beginning of a pending read cycle is delayed until after the refresh is completed.

6.2.6 RAM MODULE SCHEMATIC

Figure 6-5 provides a complete schematic drawing (6 sheets) of all circuitry on the RAM Module.

6.3 UTILIZATION: RAM MODULE

This section provides information on utilization of the 16K RAM Module.

6.3.1 INSTALLATION

In installing the RAM Module, the user must take account of:

(a) environmental extremes
(b) mounting considerations
(c) electrical connections
(d) power requirements
(e) signal requirements
(f) module selection and byte selection

Environment

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conducive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity should not exceed 90%, noncondensing.

Mounting

Avoid locating the module near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections on the board, resulting in abnormally high noise levels or outright failure of the assembly.

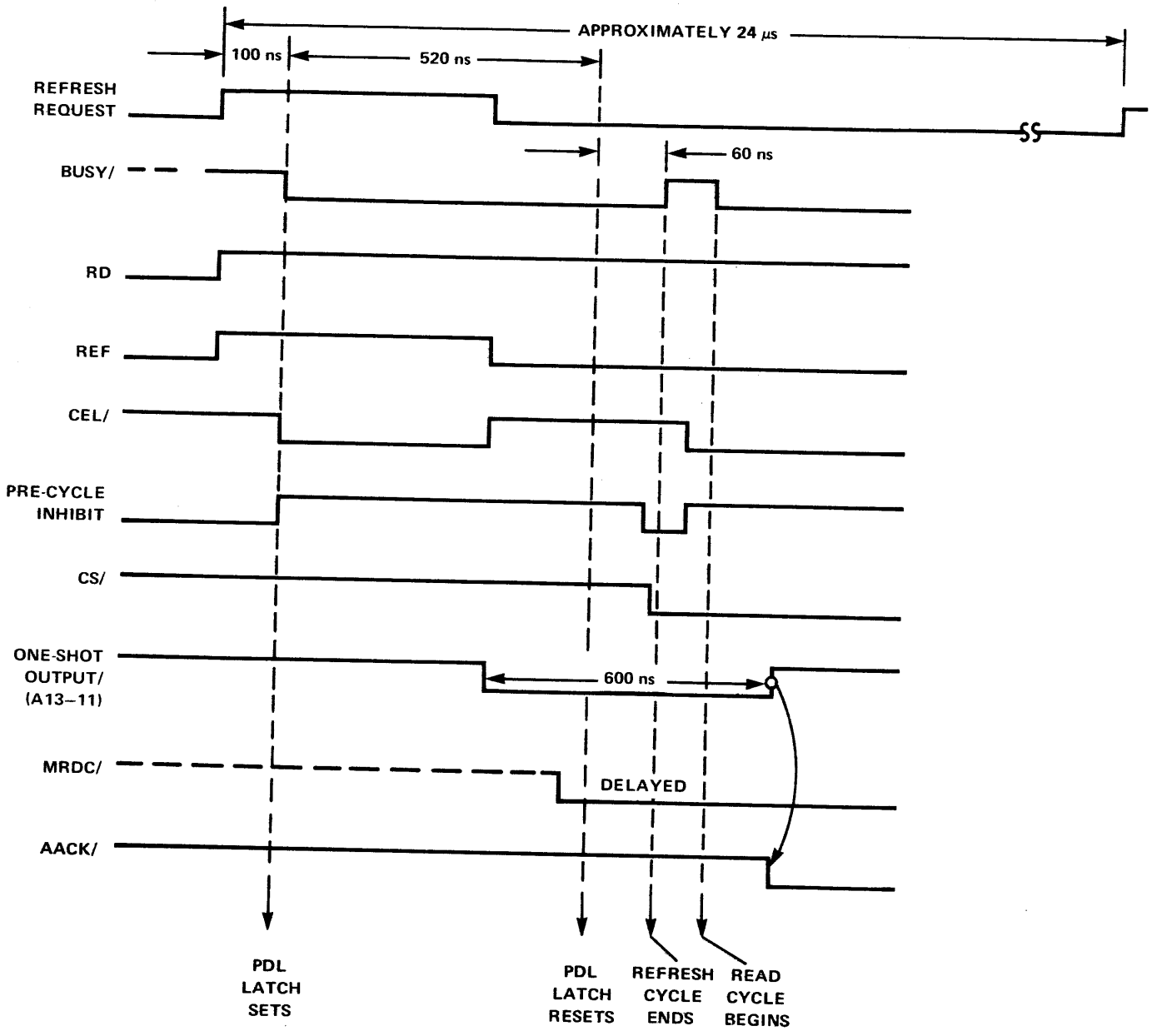


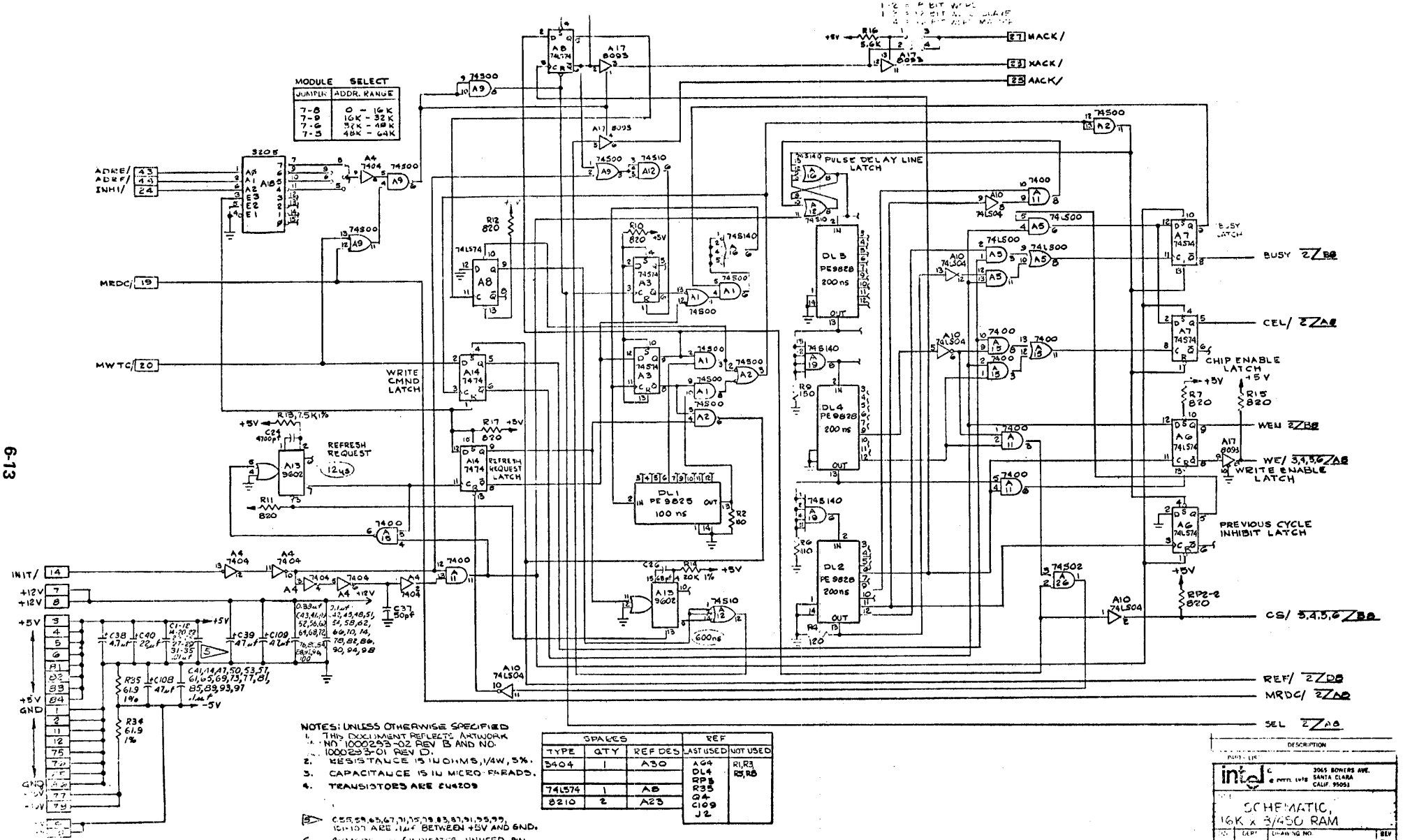
Figure 6-4. Refresh Cycle Timing (RAM Module)

Dimensions of the module are 12-in. X 6.75-in. Be sure to allow enough additional clearance to ensure adequate cooling.

The module is designed to plug directly into two standard, double-sided PC edge connectors, an 86-pin connector, and a 60-pin auxiliary connector. The connectors can serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the assembly be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the board, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the components on the module.

Electrical Connections

The 16K RAM Module communicates with the motherboard, and consequently, the rest of the system, through a standard 86-pin, double-sided



NOTES: UNLESS OTHERWISE SPECIFIED

1. THIS DOCUMENT REFLECTS ARTWORK NO 1000203-02 REV B AND NO. 100033-01 REV D.
2. RESISTANCE IS IN OHMS, 1/4W, 5%.
3. CAPACITANCE IS IN MICRO-FARADS.
4. TRANSISTORS ARE 2N4209
5. C55,59,63,67,71,75,79,83,87,91,95,99, 101-107 ARE .1μF BETWEEN +5V AND GND.
6. SYMBOL —/— INDICATES UNUSED PIN
7. SYMBOL o/o INDICATES JUMPER

SPARES			REF	
TYPE	QTY	REF DES	LAST USED	NOT USED
3404	1	A30	AG4	R1,R3
			DL4	R5,R8
			RP3	
			R35	
			Q4	
74LS74	1	A8	C109	
8210	2	A23	J2	

DESCRIPTION
intel © INTEL 1978 3065 BOWERS AVE. SANTA CLARA CALIF. 95051
SCHEMATIC, 16K x 3/450 RAM
SIZE D | DEPT 410 | DRAWING NO 2000614 | REV E

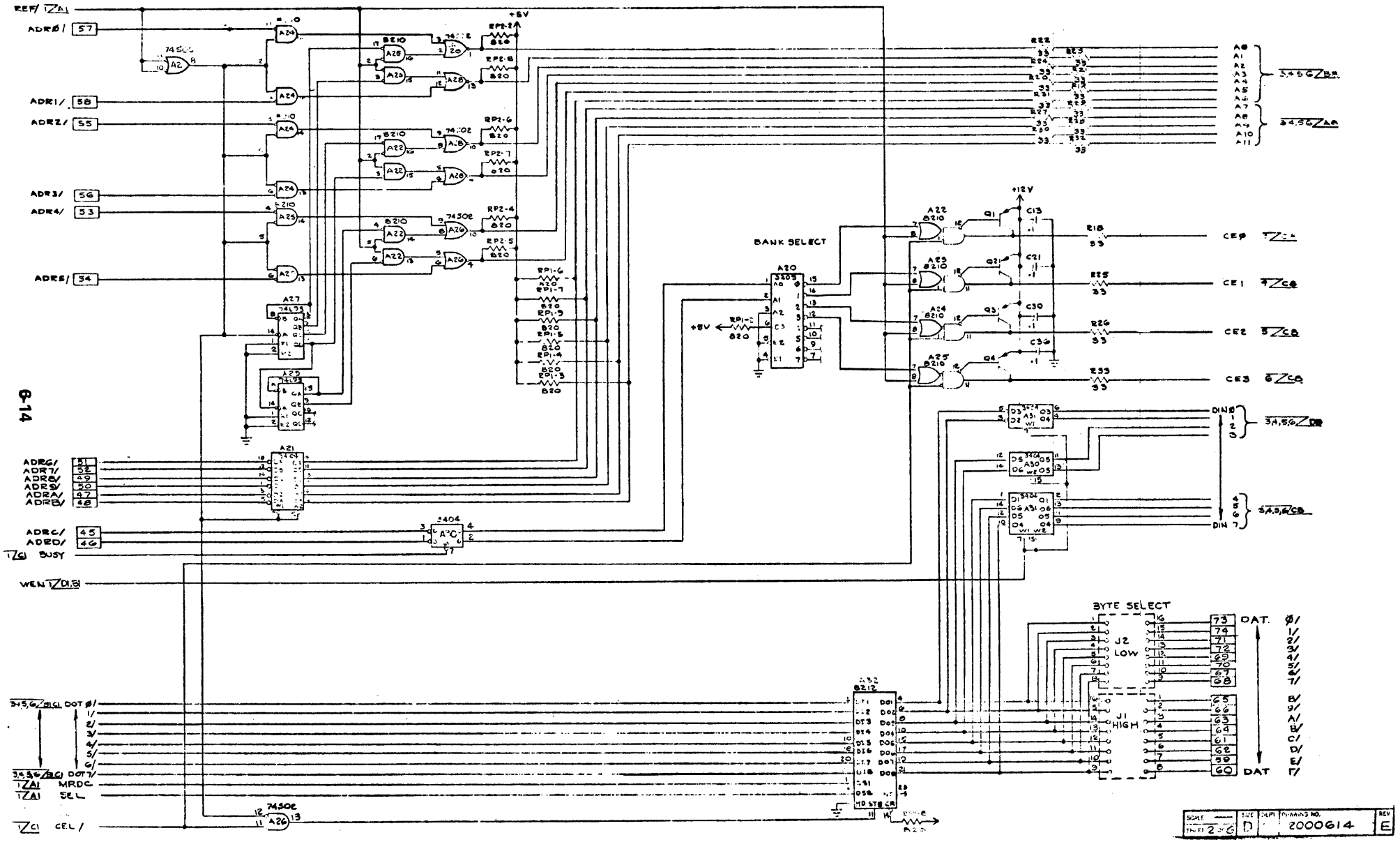


Figure 6-5. RAM Module Schematic (Sheet 2 of 6)

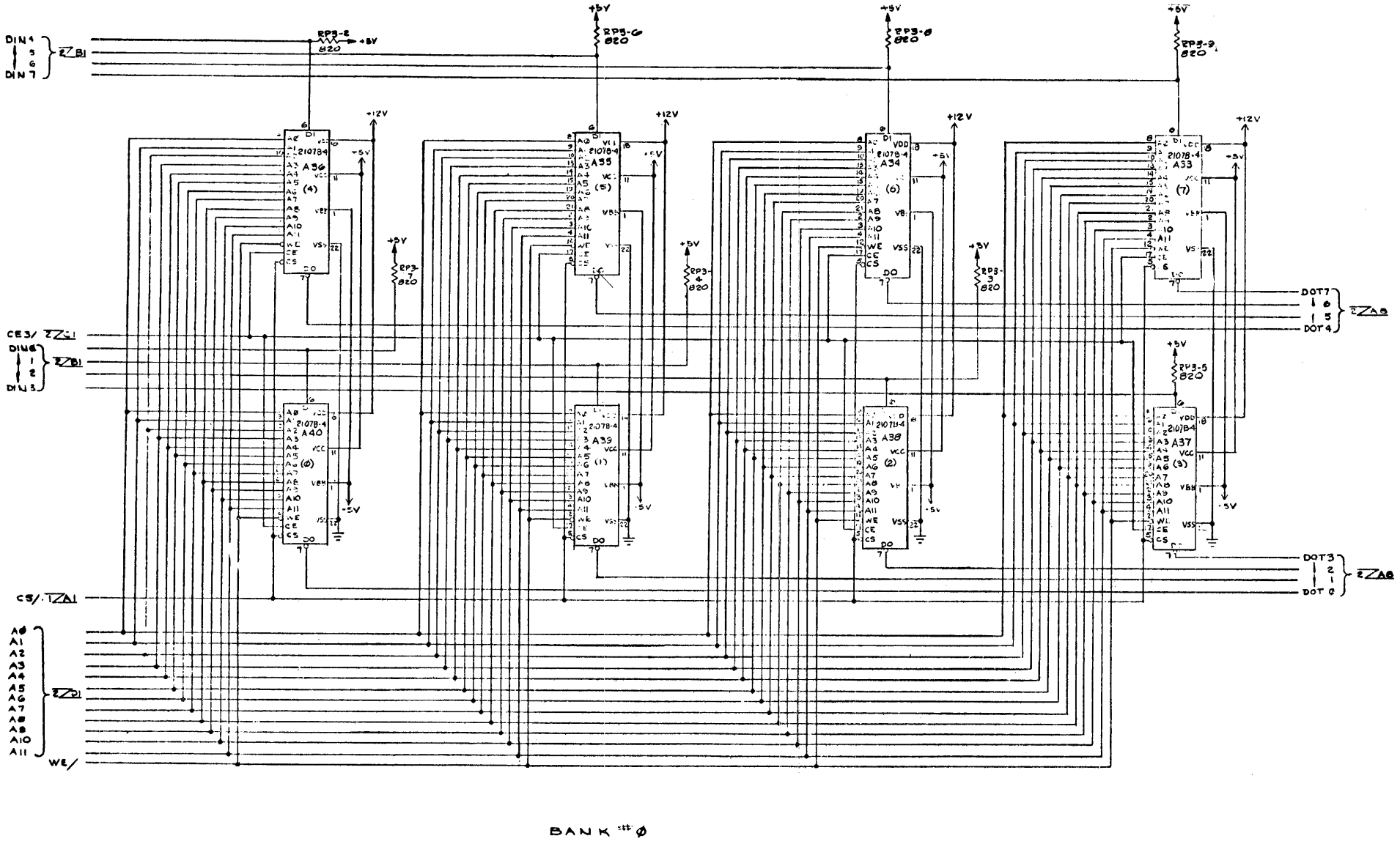
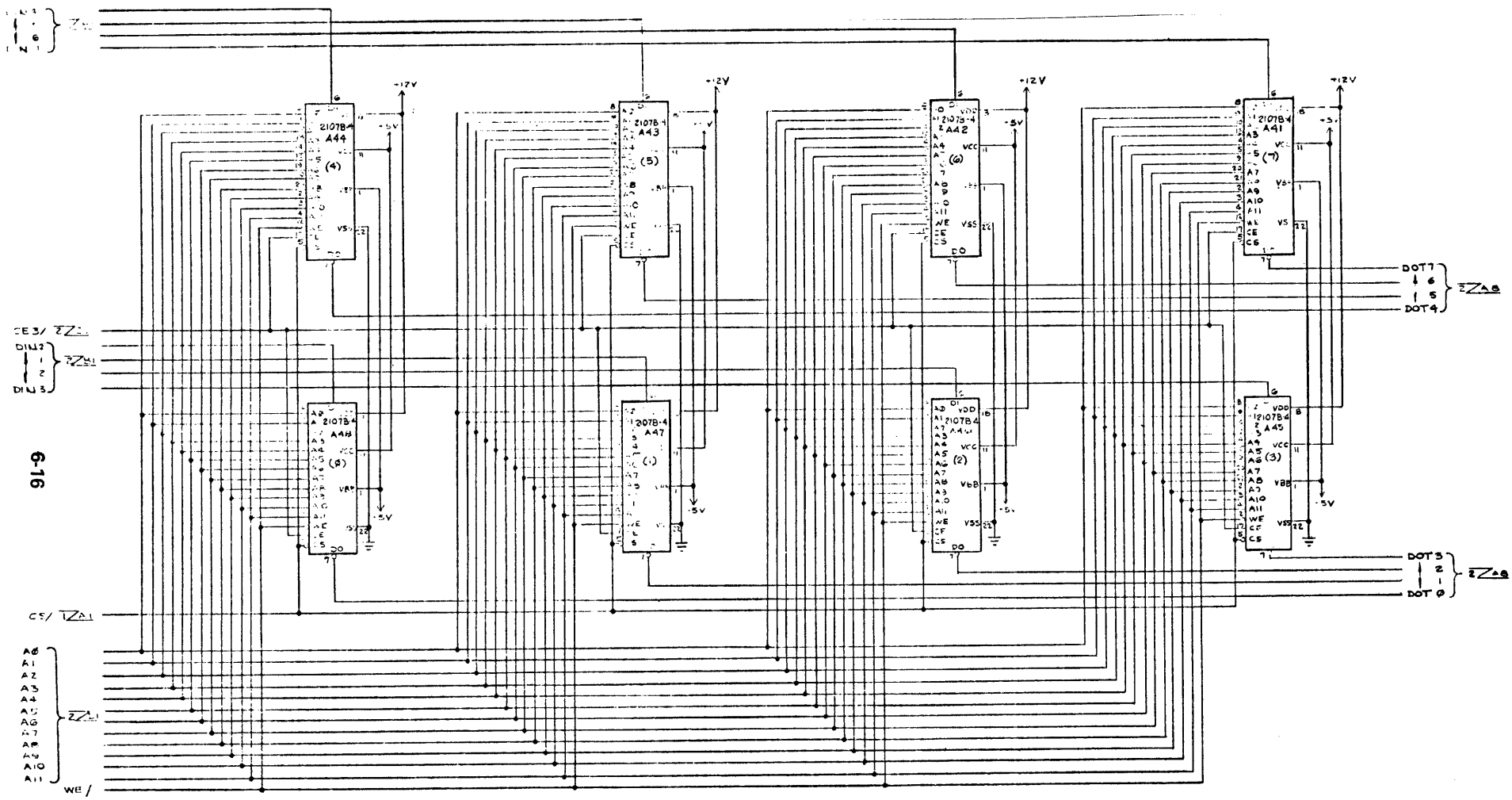
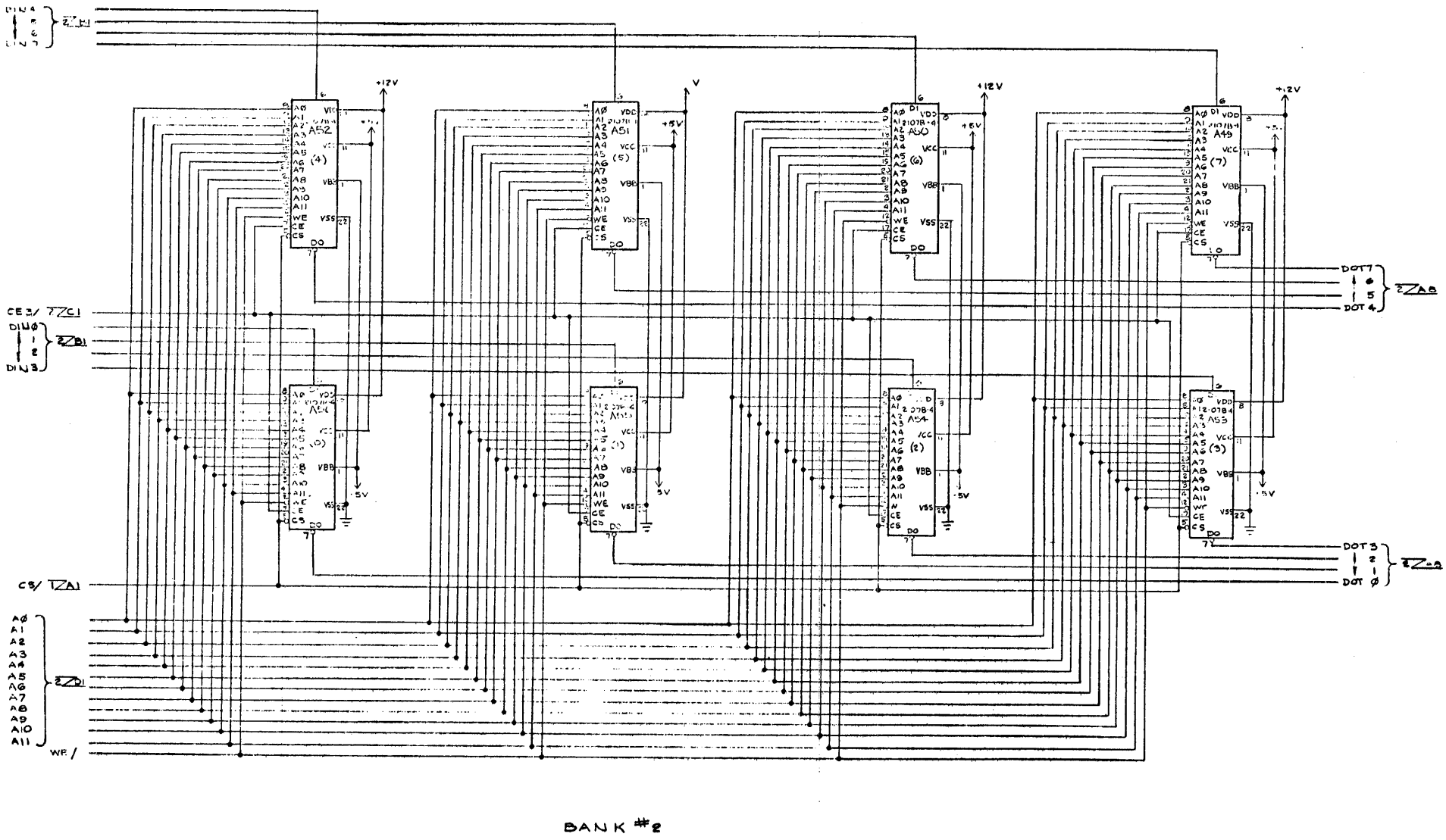


Figure 6-5. RAM Module Schematic (Sheet 3 of 6)



BANK #1

SCALE	SIZE	DRAWING NO.	REV
SHEET 4 OF 5	D	2000614	F



SCALE	SIZE	DRAWING NO.	REV
SHEET 5 OF 6	D	2000614	E

Figure 6-5. RAM Module Schematic (Sheet 5 of 6)

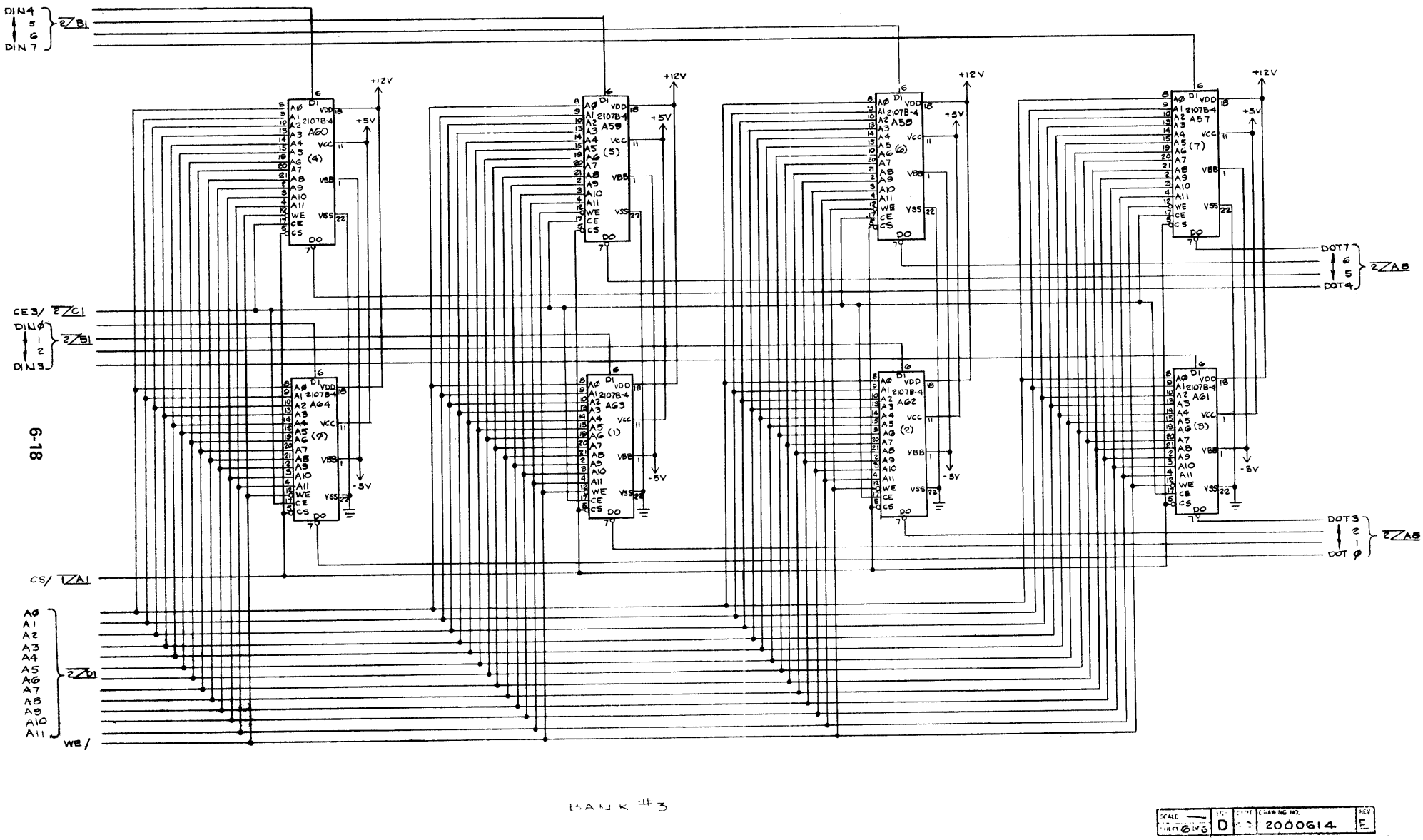


Figure 6-5. RAM Module Schematic (Sheet 6 of 6)

PC edge connector (P1), 0.156-in. contact centers, as shown in Figure 6-6. Control Data VPB01E43A-00A1 is one suitable type of connector. Pin allocations are given in Table 6-2 of Section 6.3.2. An auxiliary 60-pin, double-sided PC edge connector (P2), 0.1-in. contact centers (see Figure 6-6) is available for use at the designer's discretion.

The RAM Module requires DC power at levels of +5, −10, and +12 VDC.

Refer to the pin list, Table 6-2 of Section 6.3.2, for power connections.

Signal Requirements

All data and control functions appearing at the module edge connectors are at TTL levels. Electrical characteristics of the signal inputs and outputs, as well as power inputs, are given in Section 6.4.

Signal descriptions and connector pin allocations are given in Section 6.3.2.

Module Selection and Byte Selection

The two most significant address bits (ADRE/ and ADRF/), specify one of the four RAM Modules (or module-pairs) to be accessed. If two RAM Modules are to be paired to store 16-bit data words, both modules must have the same 2-bit module selection code. The module which is to store the low-order byte of the data word must have byte selector J2 enabled. The module which is to store the high-order byte must have byte selector J1 enabled.

Module selection codes are implemented by connecting one of the following jumper pads:

JUMPER CONNECTIONS	*SELECTION CODE	
	ADRE/	ADRF/
7-8	1	1
7-9	0	1
7-6	1	0
7-5	0	0

*NOTE: The levels on these address lines are active-low; that is, l=logical 0 and 0=logical l.

In addition, one of the paired modules must have its XACK/ line ANDed with its MACK/ line, while the other paired module has its XACK/ line disabled and the XACK/ signal diverted to drive the MACK/. These changes allow the XACK/ signal to appear on the bus (P1-23) only when both modules are ready, which is not always the case due to the asynchronous nature of refresh. To AND XACK/ and MACK/ on a module, the connection points 1-3 of the acknowledge select jumper pads should be joined. To allow the other module to drive the MACK/ line, the connection points 2-4 of the acknowledge select jumper pads should be joined.

The special acknowledge (AACK/) signal should not be used in 16-bit configurations. In 8-bit configurations where AACK/ is used, the user should disable MACK/ output by joining connection points 1-2 on the acknowledge select jumper pad (see sheet 1 of the module schematic).

6.3.2 PIN LISTS: RAM MODULE

The following section provides connector pin allocations on the 16K RAM Module. The pins and their designated signal functions for the 86-pin connector (P1) are listed in Table 6-2.

6.4 OPERATING CHARACTERISTICS: RAM MODULE

The AC and DC characteristics of all major signals that appear at the RAM Module edge connectors are provided in this section. Table 6-3 lists AC characteristics, Table 6-4 lists DC characteristics, and Figure 6-7 illustrates module timing.

6.4.1 AC CHARACTERISTICS

Detailed timing diagrams for memory operations are provided in Figure 6-7. Table 6-3 provides design limits for RAM module outputs and requirements for its inputs. These values are theoretical limits based on a worst-on-worst case analysis using vendor information and approximations where necessary. Approximations include establishing non-zero propagation delay minimums and extended delays if capacitive loading exceeds vendor ratings. In all such cases, approximations are

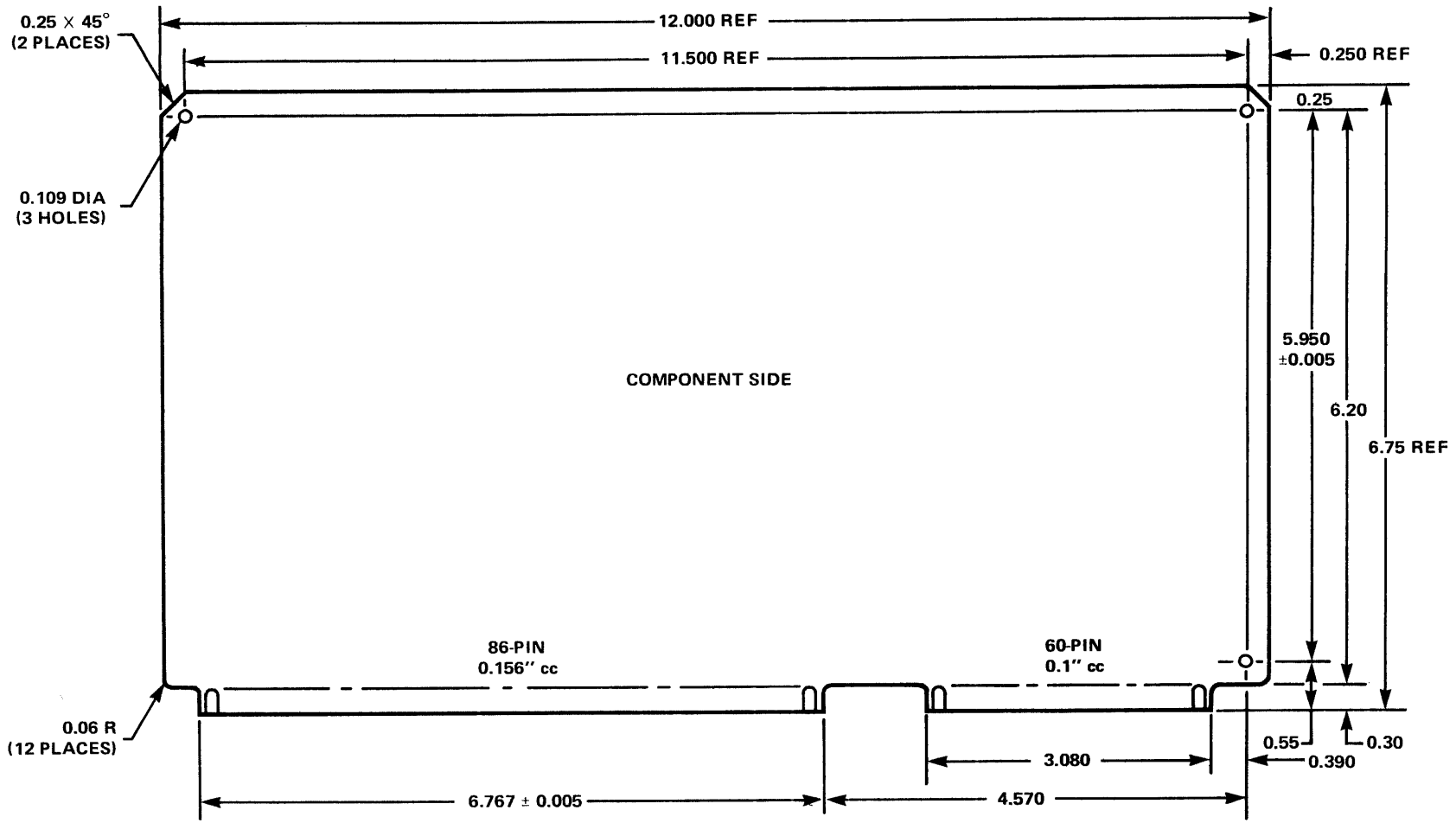


Figure 6-6. 16K RAM Module Connectors

Table 6-2

P1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	GND	Ground
2	GND	
3	+5 VDC	Power inputs
4	+5 VDC	
5	+5 VDC	
6	+5 VDC	
7	+12 VDC	
8	+12 VDC	
9		
10		
11	GND	Ground
12	GND	
13		
14	INIT/	System reset
15		
16		
17		
18		
19	MRDC/	Memory read command
20	MWTC/	Memory write command
21		
22		
23	XACK/	Acknowledge
24	INHI/	Inhibit RAM
25	AACK/	Advance acknowledge
26		
27	MACK/	Memory acknowledge
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43	ADRE/	Address bus

PIN	SIGNAL	FUNCTION
44	ADRF/	Address bus
45	ADRC/	
46	ADRD/	
47	ADRA/	
48	ADRB/	
49	ADR8/	
50	ADR9/	
51	ADR6/	
52	ADR7/	
53	ADR4/	
54	ADR5/	
55	ADR2/	
56	ADR3/	
57	ADR0/	
58	ADR1/	
59	DATE/	Data bus
60	DATF/	
61	DATC/	
62	DATD/	
63	DATA/	
64	DATB/	
65	DAT8/	
66	DAT9/	
67	DAT6/	
68	DAT7/	
69	DAT4/	
70	DAT5/	
71	DAT2/	
72	DAT3/	
73	DAT0/	
74	DAT1/	
75	GND	Ground
76	GND	
77	-10 VDC	Power inputs
78	-10 VDC	
79		
80		
81	+5 VDC	Power inputs
82	+5 VDC	
83	+5 VDC	
84	+5 VDC	
85	GND	Ground
86	GND	

Table 6-3

AC CHARACTERISTICS: 16K RAM MODULE

PARAMETER	MIN. (ns)	MAX. (ns)	DESCRIPTION	REMARKS
t_{AS}	40		Address Setup Time to Command	
t_{AH}	0		Address Hold Time from Command	
t_{C0}		450	Command to Read Data Access Time	
t_{RCY}	616	735	Read Cycle Time	
t_{80}	62	130	Command to AACK/. Time	
t_{ACK}	530	690	Command to Bus Acknowledge Time	
t_{CI}		500	Command to Write Data Setup Time	
t_{DH}	40		Write Data Hold Time	
t_{WCY}	1140	1360	Write Cycle Time	
t_{AKH}	10	34	End of Command to High Z State	
t_{80H}	7	22	End of Command to High Z State	
t_{RD}	0	735	Refresh Delay Time	Since Refresh is Asynchronous, This Time May Be Added to t_{C0}, t_{RCY}, t_{ACK}, and t_{WCY}.
t_{80D}	0	1200	AACK/. Refresh Inhibit Time.	This Time May Be Added to t_{80}.
t_{RH}	0	30	End of Command to High Z State	
t_{A80}	3	10	AACK/ Off Time From Bus ACK.	

conservative (e.g., 2 ns minimum for standard TTL, 4 ns minimum for three-state turn-offs or turn-ons). Rise and fall times are assumed to be zero unless a three-state high impedance state or open collector circuit is involved.

6.4.2 DC CHARACTERISTICS

The DC characteristics for all INTELLEC MDS Bus functions provided by this board are given in Table 6-4. They are derived from vendor specifications and calculated values if passive loading exists. Capacitance values are approximations only.

Power requirements are cited below:

		TYP	MAX
V_{CC}	+5VDC ± 5%	1.2A	1.5A
V_{DD}	+12VDC ± 5%	0.7A	1.0A
V_{BB}	−10VDC ± 5%	0.08A	0.09A

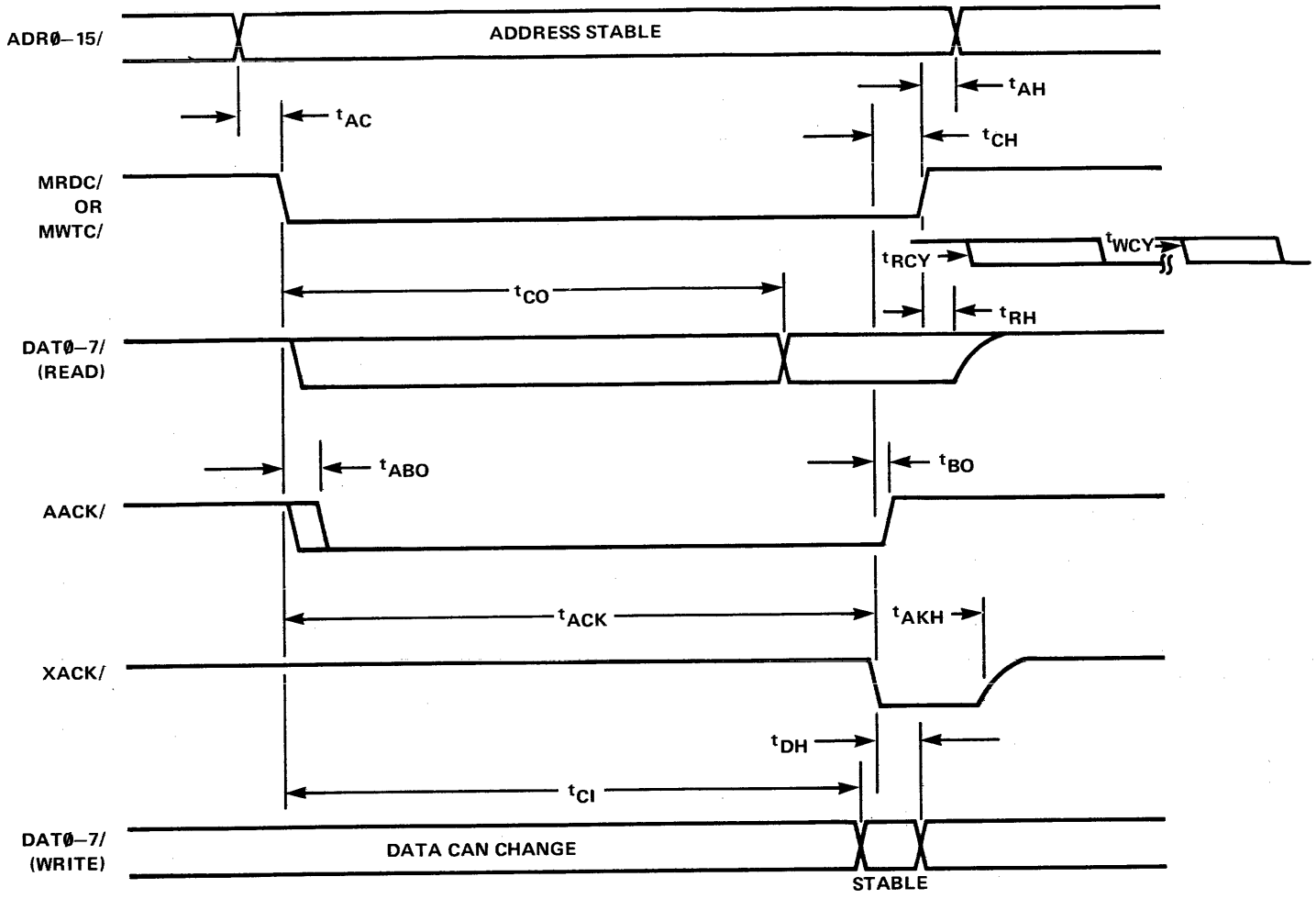


Figure 6-7. RAM Module Timing Diagram

Table 6-4

DC CHARACTERISTICS: 16K RAM MODULE

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN.	MAX.	UNIT
INIT/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		–1.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		40	μA
	C_L	Capacitive Load				pF
AACK/ XACK/	V_{OL}	Output Low Voltage	I_{OL} = 16 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = –5.2 mA	2.4		V
	C_L	Capacitive Load				pF
MACK/ (Jumper 2–4)	V_{OL}	Output Low Voltage	I_{OL} = 14.4 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = –5.2 mA	2.4		V
	C_L	Capacitive Load				pF
MACK/ (Jumper 1–3)	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		–2.4	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		40	μA
	C_L	Capacitive Load				pF
ADRØ/– ADR5/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		–0.25	mA
	I_{IH}	Input Current at High V	V_{IN} = 12.6V		10	μA
	C_L	Capacitive Load			10	pF
ADR6/– ADRF/ INHI/	V_{IL}	Input Low Voltage			0.85	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		–0.25	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		10	μA
	C_L	Capacitive Load		5	typ	pF
DATØ/– DATF/	V_{IL}	Input Low Voltage			0.85	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		–0.35	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		110	μA
	C_L	Capacitive Load		13	typ	pF
	V_{OL}	Output Low Voltage	I_{OL} = 15 mA		0.45	V

Table 6-4

DC CHARACTERISTICS: 16K RAM MODULE (continued)

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN.	MAX.	UNIT
MRDC/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		-3.0	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		90	μA
	C_L	Capacitive Load			13	pF
MWTC/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.5V		-3.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		90	μA
	C_L	Capacitive Load				pF

Chapter 7
PROM MODULE

The PROM Module has been designed to provide up to 6,144 (6K) X 8-bit words of PROM storage for 8-bit computer systems or 2,048 (2K) X 16-bit words of storage for 16-bit computer systems. Up to twenty-four 8702A erasable and electrically programmable read only memory (PROM) devices can be included on the module. Each 8702A PROM provides 256 X 8 bits of storage. Any one of the four currently available versions of the popular 8702A memory can be used with the PROM Module:

- 8702A access time = 1.0 μs
- 8702A-S614 access time = 1.5 μs
- 8702A-S314 access time = 1.7 μs
- 8702A-S714 access time = 2.5 μs

Intel's 1702A PROM's or 1302 ROM's (both pin-compatible with the 8702A) can also be used on the PROM Module, in place of the 8702A's.

The 24 PROM elements are organized into a 4K memory bank and a 2K memory bank. The user independently selects the address range for the 4K and 2K memory banks on 4K or 2K boundaries, respectively. Any address blocks within the maximum 64K range can be selected. The addresses assigned to the 2K memory bank can even coincide with those assigned to 2048 X 8-bit words in the 4K bank to implement a 2048 X 16-bit PROM storage capacity.

The PROM Module is available as an optional component within the INTELLEC MDS System, or can be obtained independently on an OEM basis. The module is implemented on a single 12-in. X 6.75-in. printed circuit board. The module requires only DC power at levels of +5 VDC and −10 VDC.

7.1 FUNCTIONAL ORGANIZATION OF THE PROM MODULE

For descriptive purposes, the PROM Module can be viewed as consisting of four functional blocks:

- Memory storage block
- Address control block
- Timing control block
- Byte selection block

These functional units are illustrated in Figure 7-1.

The *memory storage block* consists of up to twenty-four 1702A (8702A) erasable and electrically programmable-read-only-memory (PROM) elements. Each 1702A element stores 256 X 8-bit words of data (2048 bits). The 24 elements are organized into two switch-selectable banks. One bank includes 16 elements and provides 4096 (4K) X 8 bits of PROM storage. The other bank includes eight elements and provides 2048 (2K) X 8 bits of storage.

The *address control block* determines whether the 4K or 2K memory bank is to be accessed (or both for 16-bit words), as well as determining the particular location within the selected bank(s). The address control block includes three jumper pads, and two nine-position rotary switches. The combination of jumper connection and switch settings determine how the 16-bit address from the CPU is to be decoded.

The *timing control block* is responsible for generating the acknowledge signals (AACK/ and XACK/), the inhibit RAM signal (INH1/) and the DATA ENABLE/ signal. The AACK/ and XACK/ signals are returned to the CPU; INH1/ prevents RAM devices from responding to memory addresses intended for the PROM Module; and the DATA ENABLE/ signal strobes the data words (8 or 16-bit) through the byte selection logic and onto the bus. Because several versions of the 8702A PROM are available (each version having a different access time), the timing control block includes an access time select switch which defines delay times for the AACK/ and XACK/ acknowledge signals.

The *byte selection block* controls the flow of data to the data bus via an eight-pair jumper pad (J4).

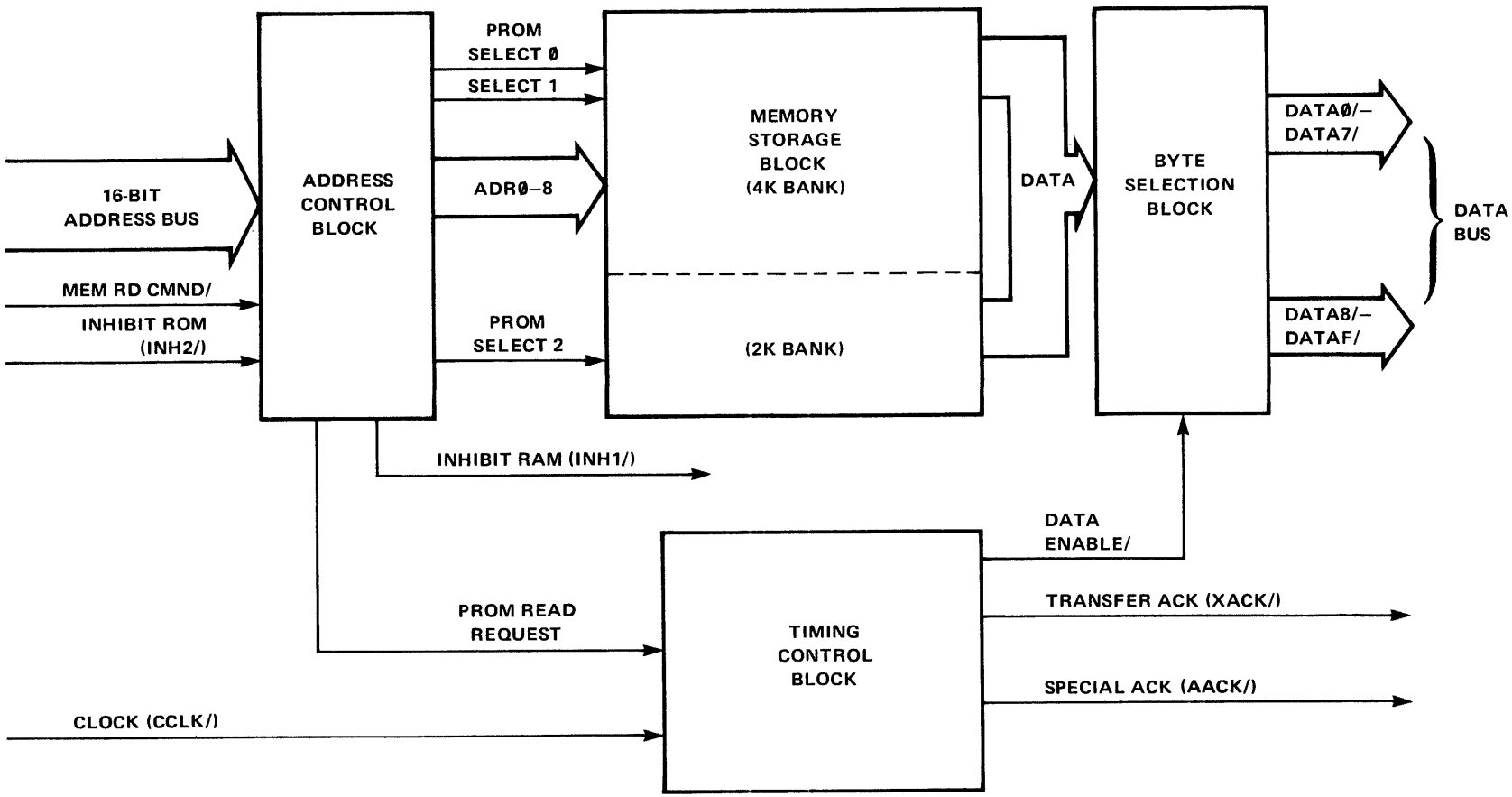


Figure 7-1. PROM Module Functional Block Diagram

The byte selection block can be configured to allow 16-bit data words; jumper pad J4 is removed. In such a case, both the 4K and 2K memory banks are enabled in the address control block, though only 2048 bytes of the 4K bank will be used. The byte selection logic, in the absence of J4, would route the data byte from the 4K bank to the low-order data bus lines (DATAØ/–DAT7/) and would route the data byte from the 2K bank to the high-order data bus lines (DATA8/–DATAF/). In 8-bit systems, the J4 jumper pad must be present to provide a data path from the 2K memory bank to the low-order data bus lines (DATAØ/–DATA7).

7.2 PROM MODULE: THEORY OF OPERATION

In this section, we provide a detailed theory of operation description for the PROM Module. The schematic (4 sheets) for the PROM Module is provided in Figure 7-6, located in Section 7.2.4.

7.2.1 PHYSICAL MEMORY IMPLEMENTATION

The actual memory on the PROM Module consists of up to twenty-four 8702A programmable-read-only-memory (PROM) elements. Each 8702A element has a 256 X 8-bit capacity. The 24 PROM elements are partitioned into two memory banks. One bank includes 4096 (4K) X 8 bits of storage (16 PROM elements), while the other bank includes 2048 (2K) X 8 bits (8 PROM elements). The address ranges for the two banks are switch-selectable as described in Section 7.2.2. In addition, the 2K bank can be used with 2048 words in the 4K bank to implement 2K X 16-bit word storage; as previously described.

The 8702A PROM's are shown on sheets 2, 3 and 4 of the module schematic, Figure 7-6.

7.2.2 MEMORY ADDRESS DECODING

The address control block is responsible for decoding the 16-bit address output by the CPU during all PROM read operations. The address control logic includes three jumper pads and two rotary switches that determine the specific address space to be occupied by the PROM elements. The majority of the address control logic is shown on sheet 1 of the module schematic, Figure 7-6.

The 16 address bits are received by the PROM Module at pins P1-43 through P1-58 and buffered by 3404 inverting buffer circuits. The eight least significant address lines (ADRØ/–ADR7/) are applied to the address inputs of the 8702A PROM elements (AØ–A7). These address lines uniquely identify one of 256 X 8-bit words in each PROM.

Address lines ADR8/, ADR9/, and ADRA/, each feed the address inputs of three 3205 decoders (shown on sheets 2, 3 and 4 of the module schematic). Each decoder is associated with eight PROM elements. The decoders are enabled by one of the three PROM SELECT signals (derived from the five most significant address bits as described below). Each of the eight decoder outputs is applied to the chip select input on one of the eight associated PROMs. During a PROM read operation, only one of the outputs (specified by ADR8/, ADR9/, and ADRA/) on one of the decoders (specified by ADRB/–ADRF/) will be active. Thus, only that PROM element which is specified by the eight high-order address bits is enabled.

The five most significant address lines (ADRB/–ADRF/) are directed to the address selection network, where the PROM SELECT signals are generated. Address line ADRB/ is applied (in either an active-low or active-high form) to one of two inputs on the three 7402 negative-AND gates that actually generate the PROM SELECT signals. The other input to each of these gates is supplied by one of the two rotary switches (S3 and S4). The nine-position rotary switches are tied back to the outputs of two 3205 decoders. Address lines ADRC/, ADRD/, and ADRE/ feed the address inputs of these two decoders. The two decoders are enabled by address line ADRF/. The particular level (high or low) on ADRF/ that enables the decoders is dependent on the setting of the X2 and Y2 jumper pads (pads 7-8-9 and 10-11-12, respectively). A rotary switch output, then, will be active (low) only if its associated decoder is enabled by ADRF/ (as specified by X2 of Y2) and the value on address lines ADRC/–ADRE/ matches the setting of the switch (position 9 means the switch is off).

To generate the PROM SELECT Ø signal, the output of switch S4 must be low (active) but ADRB/ must be high (inactive). To generate PROM SELECT 1, the output of switch S4 must be low (active), and ADRB/ must be low (inactive). To generate PROM SELECT 2, the output of switch S3 must be active and the 2K address select jumper (pad 1-2-3) must be connected such that the input to A41-8 is low.

PROM SELECT Ø enables the least significant 2048 X 8-bit words of the 4K memory bank (sheet 2 of the schematic). PROM SELECT 1 enables the most significant 2048 X 8-bit words of the 4K memory bank (sheet 3 of the schematic). PROM SELECT 2 enables the 2K memory bank.

To summarize: The five most significant address bits (ADRB/–ADRF/) specify one of three 2048 word sections (eight PROM elements per section) within the PROM Module. The particular 2048 word section which is enabled is indicated by one of the three PROM SELECT signals. Address bits ADR8/, ADR9/, and ADRA/, in turn, enable one of the eight PROMs in the selected 2048 word section. Finally, the eight least significant address bits (ADRØ/–ADR7/) specify one of 256 X 8-bit words within the selected PROM element.

As we mentioned above, the actual 2048 word section which is selected by decoding the five high-order address bits is dependent on a number of variables. That is, various jumper connections and switch settings determine the actual addresses that a particular PROM element will respond to. In essence, the user assigns a particular set of addresses to each of the three 2048 word sections (8 PROM elements per section) on the module.

The addresses for the 2K memory bank are determined by the following connections and switch settings:

- 2K address select jumper pad (1-2-3)
- Y2 decoder enable jumper pad (10-11-12)
- Rotary switch (S3) referred to as Y1

The addresses for the 4K memory bank are determined by these connections and switch settings:

- X2 decoder enable jumper pad (7-8-9)
- Rotary switch (S4), referred to as X1

The following technique can be used for address selection:

1. If the module is being used in an 8-bit configuration, the address range of the *4K bank* is-

   XØØØ to XFFF (base 16)

   where $X = X_1 + X_2$ (hexadecimal addition),

   X_1 is determined by switch X_1
   (values 0 to 7 as detailed on the PCB), and

   X_2 is determined by jumper X_2
   (values 0 or 8 as detailed on the PCB).

   The address range of the *2K bank* is:

   YØØØ to Y7FF
   or Y8ØØ to YFFF (base 16)

   where $Y = Y_1 + Y_2$

   Y_1 and Y_2 are selected in a manner similar to X_1 and X_2, described above.

   The second most significant hexadecimal digit in the 2K bank address range is determined by jumper 1-2-3. One position yields YØØØ, the other Y8ØØ.

2. If the module is being used in a 16-bit configuration, connector J4 must be removed. Half of the 4K bank is used for the upper byte of the 16-bit words, while the 2K bank is used for the lower byte. X and Y are selected as described above. In 16-bit configurations, however, X and Y must be set equal.

Table 7-1 lists the addresses that result from the various combinations of jumper connections and switch settings.

Note in Table 7-1 that the address range for the 2K memory bank can be different than that for the 4K memory bank or it can respond to the same addresses as one of the two 2048 word sections within the 4K memory bank. When the 2K and 4K memory banks have mutually exclusive address ranges, the PROM Module provides 6,144 (6K) X 8-bit words of storage. When the address range for the 2K module is the same as the address range for

Table 7-1

ADDRESS SWITCH POSITIONS

ADDRESS RANGE (HEX)	4K BANK			2K BANK			
	X1* [SWITCH S4]	X2 [JUMPER 7-8-9]	PROM** CHIP LOCATIONS	Y1* [SWITCH S3]	Y2 [JUMPER 10-11-12]	2K ADDRESS SELECT [JUMPER 1-2-3]	PROM** CHIP LOCATIONS
0000–07FF	1	8–9	A1–A8	1	11–12	2–1	A23–A30
0800–0FFF	1		A12–A19	1		2–3	
1000–17FF	2		A1–A8	2		2–1	
1800–1FFF	2		A12–A19	2		2–3	
2000–27FF	3		A1–A8	3		2–1	
2800–2FFF	3		A12–A19	3		2–3	
3000–37FF	4		A1–A8	4		2–1	
3800–3FFF	4		A12–A19	4		2–3	
4000–47FF	5		A1–A8	5		2–1	
4800–4FFF	5		A12–A19	5		2–3	
5000–57FF	6		A1–A8	6		2–1	
5800–5FFF	6		A12–A19	6		2–3	
6000–67FF	7		A1–A8	7		2–1	
6800–6FFF	7		A12–A19	7		2–3	
7000–77FF	8		A1–A8	8		2–1	
7800–7FFF	8	8–9	A12–A19	8	11–12	2–3	
8000–87FF	1	8–7	A1–A8	1	11–10	2–1	
8800–8FFF	1		A12–A19	1		2–3	
9000–97FF	2		A1–A8	2		2–1	
9800–9FFF	2		A12–A19	2		2–3	
A000–A7FF	3		A1–A8	3		2–1	
A800–AFFF	3		A12–A19	3		2–3	
B000–B7FF	4		A1–A8	4		2–1	
B800–BFFF	4		A12–A19	4		2–3	
C000–C7FF	5		A1–A8	5		2–1	
C800–CFFF	5		A12–A19	5		2–3	
D000–D7FF	6		A1–A8	6		2–1	
D800–DFFF	6		A12–A19	6		2–3	
E000–E7FF	7		A1–A8	7		2–1	
E800–EFFF	7		A12–A19	7		2–3	
F000–F7FF	8		A1–A8	8		2–1	
F800–FFFF	8	8–7	A12–A19	8	11–10	2–3	A23–A30

*Position 9 is OFF.

**NOTE: PROM chip locations A1–A8 are enabled by PROM SELECTØ, locations A12–A19 by PROM SELECT1 and locations A23–30 by PROM SELECT2.

2048 words within the 4K bank, the module will provide 2048 (2K) X *16-bit* words of storage. In this configuration, only 16 PROM elements would be resident on the module.

The address control block also includes a PROM resident select switch which prevents an attempted access to a non-existent portion of PROM storage from being executed (e.g., if all 24 PROM positions on the module are not being used). One of the three PROM SELECT signals is generated whenever the PROM Module is being accessed. In addition to enabling the selected PROM, the PROM SELECT signals are each applied to one of three 7403 NAND gates. The other input on each of these 7403 sections is tied to +5 VDC (through a resistor) and to one of three poles on the PROM resident select switch (S1). If a particular PROM SELECT signal is true and the associated pole on the S1 switch is open (indicating that the addressed 2048 word section is present and operational), the 7403 gate is activated. Table 7-2 lists the settings on the PROM resident switch (S1) that enable the three 2048 word blocks on the module.

The outputs of the three 7403 gates are wire-ORed together, then inverted. This inverted result is inverted again and asserted at connector pin P1-24 as the inhibit RAM signal (INH1/). INH1/ prevents a RAM device from responding to a memory address intended for the PROM Module.

The inverted, ORed output of the three 7403 gates is also applied to one input of a 7408 AND gate. The other 7408 input is furnished by the result of ANDing the memory read command (MRDC) and the active-low inhibit ROM signal (INH1/). The output of the 7408 gate constitutes the PROM READ REQUEST signal. PROM READ REQUEST is made available to the timing control block (see Section 7.2.3).

7.2.3 PROM TIMING CONTROL

The timing control logic is responsible for acknowledging the PROM read operation and enabling the byte selection logic, at the proper times. Because various 8702A PROMs with different access times are currently available, the timing control logic has been designed to selectively coordinate overall PROM Module timing with the speed of the PROM devices actually being used. The timing control block includes an eight-pole access timer select switch which allows the user to specify module timing as a function of the PROM type.

PROM READ REQUEST, which is normally held low, goes high when a PROM is to be accessed (see Section 7.2.2). It is inverted and applied to the byte selection logic under the mnemonic DATA ENABLE/. DATA ENABLE/ enables the eight (or 16) 8098 hex inverters that drive the data bus (see sheet 4 of the module schematic). The active-high level on PROM READ REQUEST is also applied to the D-input of a 7474 flip-flop (shown at A22-12). This high level also enables two other 7474 sections, the XACK/ and AACK/ latches, which had been held in the pre-reset state by the low level on PROM READ REQUEST. The command clock signal, CCLK/ (received at pin P1-31), clocks all three 7474 flip-flops, though only the first 7474 latch (output at A22-9) will toggle to the set state at this time. The Q output of this set latch (A22-9) presents a high level to the active-low load (LD) inputs of two 74161 synchronous, 4-bit counters. The two counters are wired together as a single 8-bit counter (i.e., the carry output of the first counter enables the second counter). This 8-bit counter scheme now increments its output value once for each CCLK/ pulse received, beginning with the pre-loaded value. The pre-loaded value is determined by the setting of the access timer select switch (S2). As we mentioned above, the S2 switch setting is dependent on the type and speed of 8702A PROM device actually being used. Table 7-3 provides the proper swtich settings for each of the four 8702A PROM types.

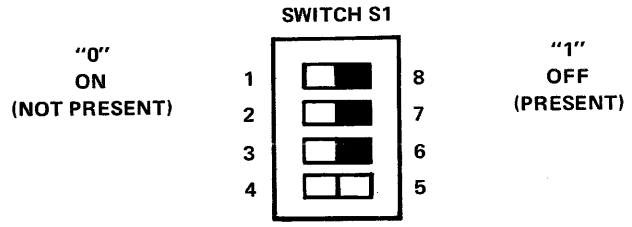
The QC output of the top 74161 counter is applied to one input of a 7408 AND gate. The QB output of the bottom 74161 counter feeds the D-input of the AACK/ latch. On the next positive-going edge of CCLK/ after QB goes high, the AACK/ latch is clocked to the set state. The low $\overline{Q}$ output of this latch is then applied to its pre-set input, thus maintaining its set condition through subsequent occurrences of CCLK/. The high Q output is inverted and driven through connector pin P1-25 as the special acknowledge signal, AACK/. The Q output is also applied to the 7408 AND gate with QC from the first counter. The output of this 7408 gate, in turn, feeds the D-input of the XACK/ latch. The next positive-going edge of CCLK/ after the 7408

Table 7-2

PROM RESIDENT SELECT SWITCH (S1) SETTINGS

MEMORY BANK	PROM CHIP LOCATIONS	S1 SWITCH POLES	SWITCH SETTINGS FOR 8702 INSTALLATION*	
			NOT PRESENT	PRESENT
2K	A23–A30 (PROM SELECT2)	1,8	0	1
4K	A1–A8 (PROM SELECT0)	2,7	0	1
	A12–A19 (PROM SELECT1)	3,6	0	1

*0=OFF, closed circuit between +5V and ground
1=ON, open circuit between +5V and ground



NOTE: POLES 4, 5 NOT USED.

SETTING SHOWN INDICATES ALL THREE 2048 WORD BLOCKS ARE PRESENT AND OPERATIONAL.

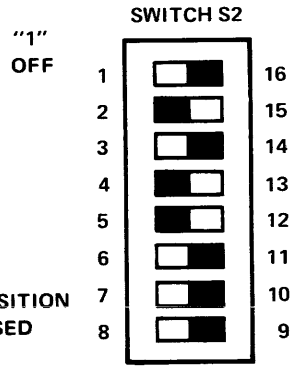
NOTE: NUMBERING REFERS TO DIP PIN DESIGNATIONS RUNNING COUNTERCLOCKWISE FROM THE UPPER LEFT CORNER OF THE SWITCH MODULE. IGNORE VENDOR DESIGNATIONS.

Table 7-3

ACCESS TIMER SELECT SWITCH (S2) SETTINGS

DEVICE TYPE	ACCESS TIME	SWITCH SETTINGS (S2)*							
		1,16	2,15	3,14	4,13	5,12	6,11	7,10	8,9
8702A	1.0 μs	0	1	0	1	1	0	0	0
8702A–S614	1.5 μs	1	0	1	0	1	0	0	0
8702A–S314	1.7 μs	0	0	1	0	1	0	0	0
8702A–S714	2.5 μs	1	1	0	1	0	0	0	0

*0=ON, closed circuit between +5V and ground
1=OFF, open circuit between +5V and ground



NOTE: DARK SIDE REPRESENTS POSITION OF SLIDER OR THE DEPRESSED SIDE OF A ROCKER SWITCH.

NOTE: NUMBERING REFERS TO DIP PIN DESIGNATIONS RUNNING COUNTERCLOCKWISE IN ASCENDING ORDER FROM THE UPPER LEFT-HAND CORNER OF THE SWITCH MODULE. VENDOR LABELING ON THE SWITCH SHOULD BE IGNORED.

SETTING SHOWN IS FOR 8702A DEVICE (ACCESS TIME = 1.0 μs

goes active clocks this latch set. The $\overline{Q}$ output is applied to the pre-set input locking the latch in the set state. The Q output is inverted and driven through connector pin P1-23 as the transfer acknowledge signal, XACK/. When DATA ENABLE/ (described above) goes false at the end of the memory read cycle, the two 8098 circuits which drive XACK/ and AACK/ are disabled.

Timing for XACK/ and AACK/ for each of the four 8702A PROM types is provided in Figures 7-2 through 7-5.

7.2.4 PROM MODULE SCHEMATIC

Figure 7-6 provides a complete schematic drawing (4 sheets) of all circuitry on the PROM Module.

7.3 UTILIZATION: PROM MODULE

This section provides information on utilization of the PROM Module.

7.3.1 INSTALLATION

In installing the PROM Module, the user must take account of:

(a) environmental extremes
(b) mounting considerations
(c) electrical connections
(d) power requirements
(e) signal requirements
(f) address assignments
(g) access timer selection
(h) byte selection

Environment

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° and 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conducive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity should not exceed 90%, noncondensing.

Mounting

Avoid locating the module near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections on the board, resulting in abnormally high noise levels or outright failure of the assembly.

Dimensions of the module are 12-in. X 6.75-in. Be sure to allow enough additional clearance to ensure adequate cooling.

The module is designed to plug directly into two standard, double-sided PC edge connectors; an 86-pin connector and a 60-pin auxiliary connector. The connectors can serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the assembly be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the board, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the components on the module.

Electrical Connections

The PROM Module communicates with the motherboard and, consequently, the rest of the system, through a standard 86-pin, double-sided PC edge connector (P1), 0.156-in. contact centers, as shown in Figure 7-7. Control Data VPB01E43A00A1 is one suitable type of connector. Pin allocations on this connector are given in Table 7-4 of Section 7.3.2. The module can also communicate with other modules in the system, through the auxiliary 60-pin, double-sided PC edge connector (P2), 0.1-in. contact centers (see Figure 7-7). Pin allocations for this connector (primarily used for test points) are listed in Table 7-5.

The PROM Module requires DC power at levels of +5 VDC and −10 VDC.

Refer to the pin lists in Tables 7-4 and 7-5 of Section 7.3.2 for power connections.

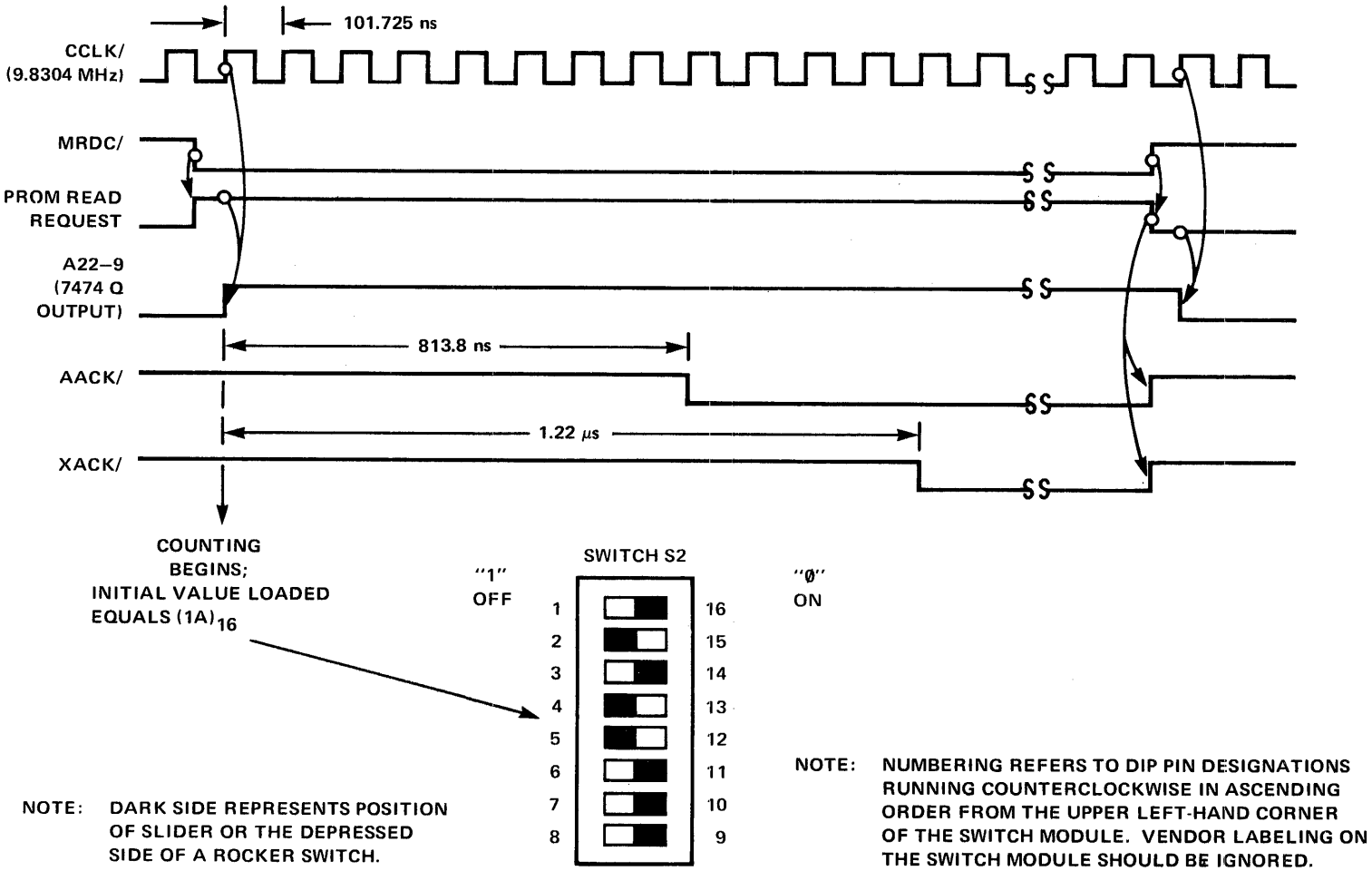


Figure 7-2. PROM Module Timing for 8702A PROMs (Access Time = 1.0 μs)

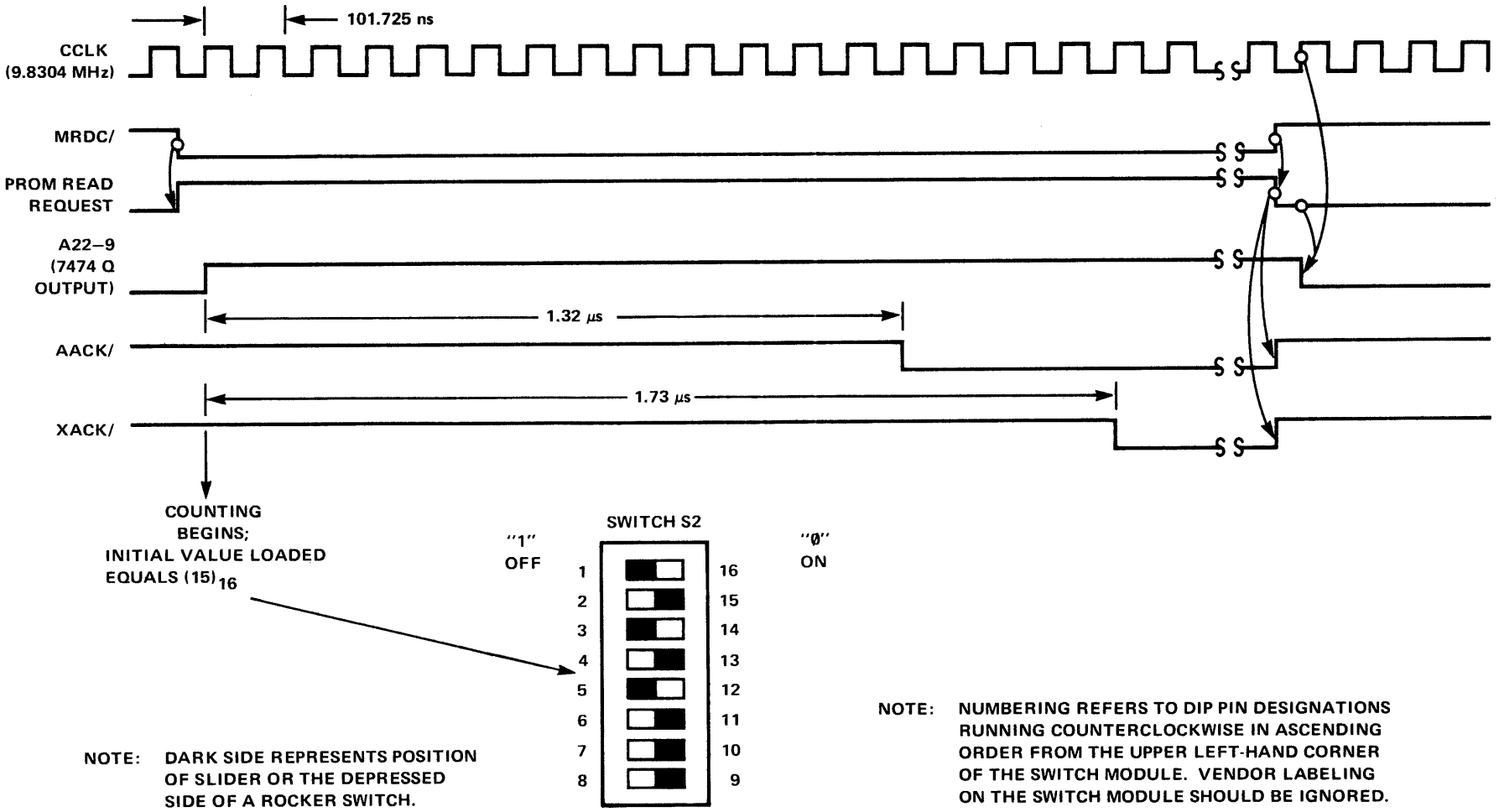


Figure 7-3. PROM Module Timing for 8702A-S614 PROMs (Access Time = 1.5 μs)

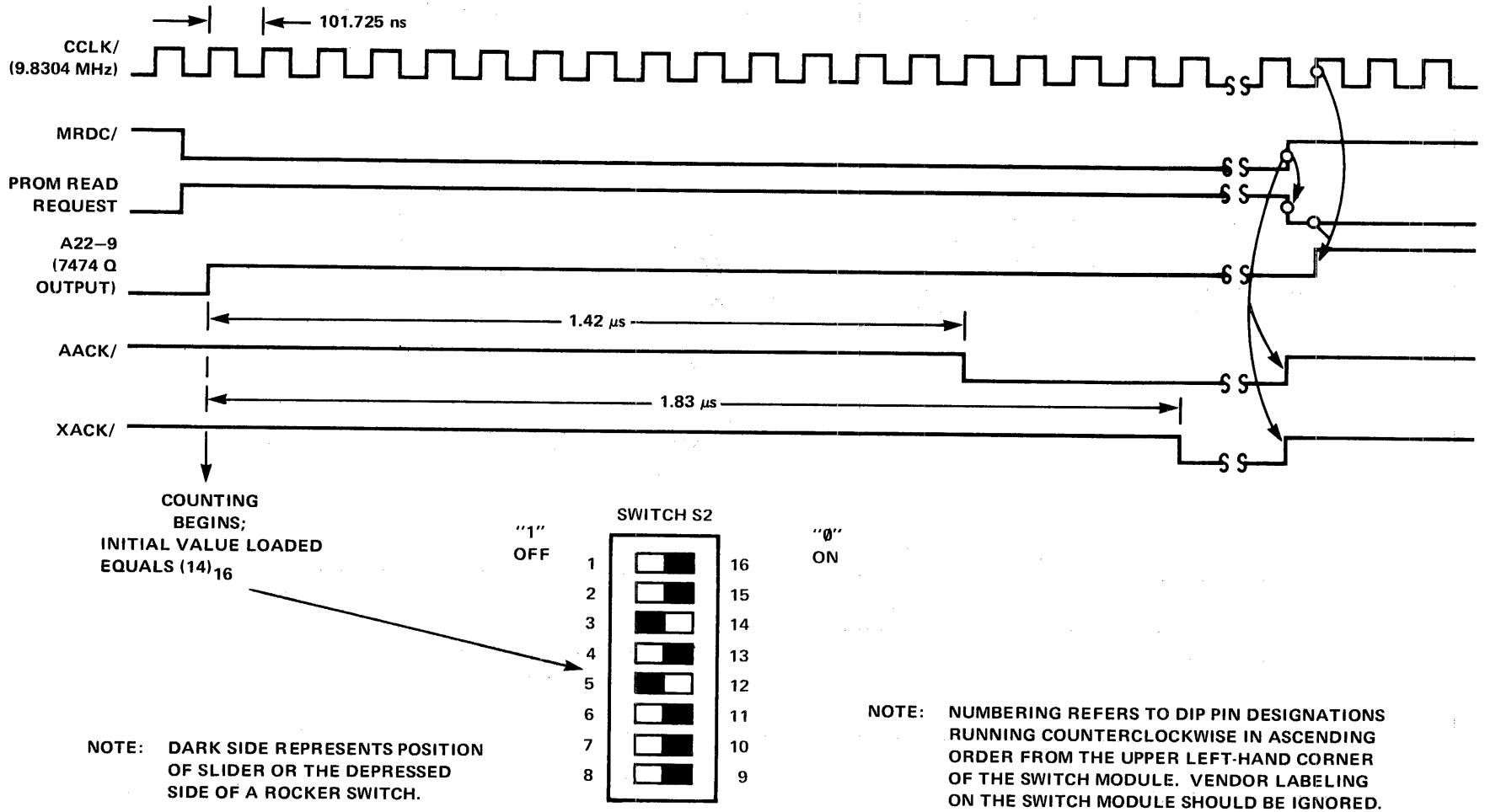


Figure 7-4. PROM Module Timing for 8702A-S314 PROMs (Access Time = 1.7 μs)

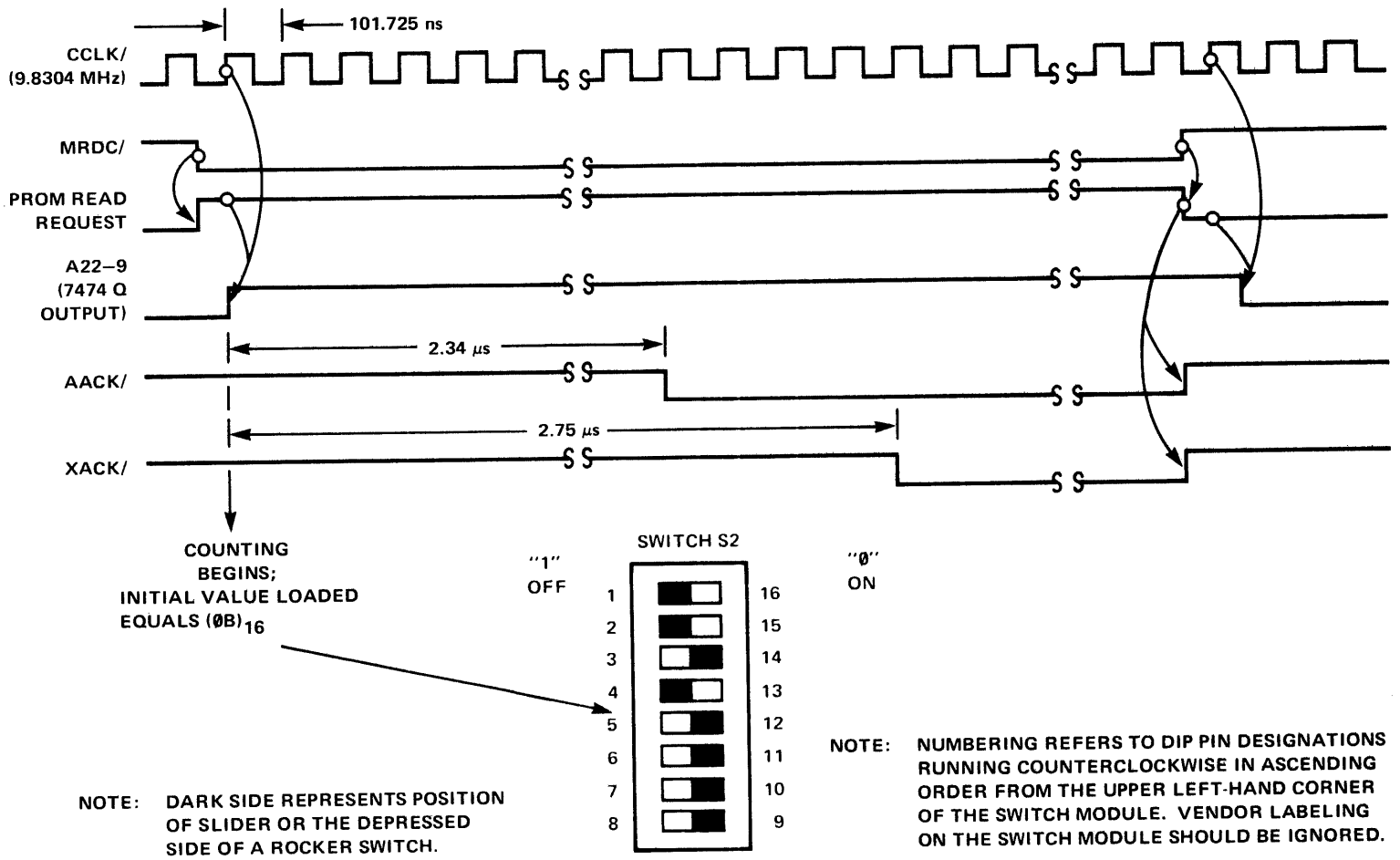
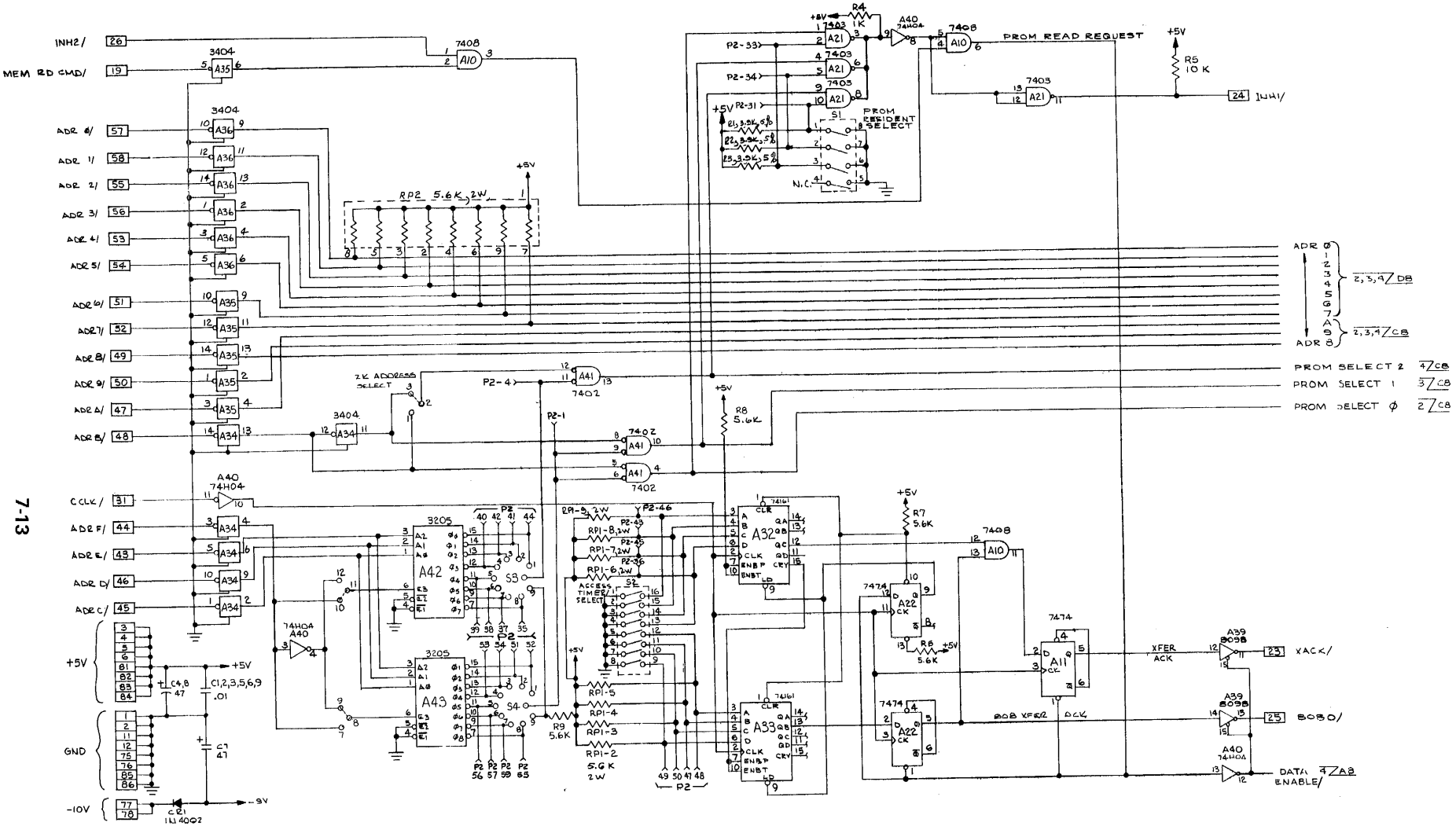


Figure 7-5. PROM Module Timing for 8702A-S714 PROMs (Access Time = 2.5 μs)



NOTE : UNLESS OTHERWISE SPECIFIED

1. ARTWORK REVISION LEVEL IS REV A
2. RESISTOR VALUES ARE IN OHMS, ± 10%, 1/4W.
3. CAPACITOR VALUES ARE IN MICROFARADS.
4. J4 JUMPERS INSTALLED WHEN MODULE USED AS -01, J1,2,3 AND 5 USED FOR TEST PURPOSES ONLY.
5. ALL 1702A PROMS ARE OPTIONAL ITEMS AND ARE SHOWN ON SCHEMATIC FOR CIRCUIT CONTINUITY ONLY. (REF 6K X 8 A1-A8, A12-A19, A23-A30) (REF 2K X 16 A12-A19, A23-A30)

Figure 7-6. PROM Module Schematic (Sheet 1 of 5)

intel © INTEL 1975	3065 BOWERS AVE. SANTA CLARA CALIF. 95051		
TITLE SCHEMATIC, PROM MEMORY MODULE			
SIZE D	DEPT 410	DRAWING NO. 2000315	REV B

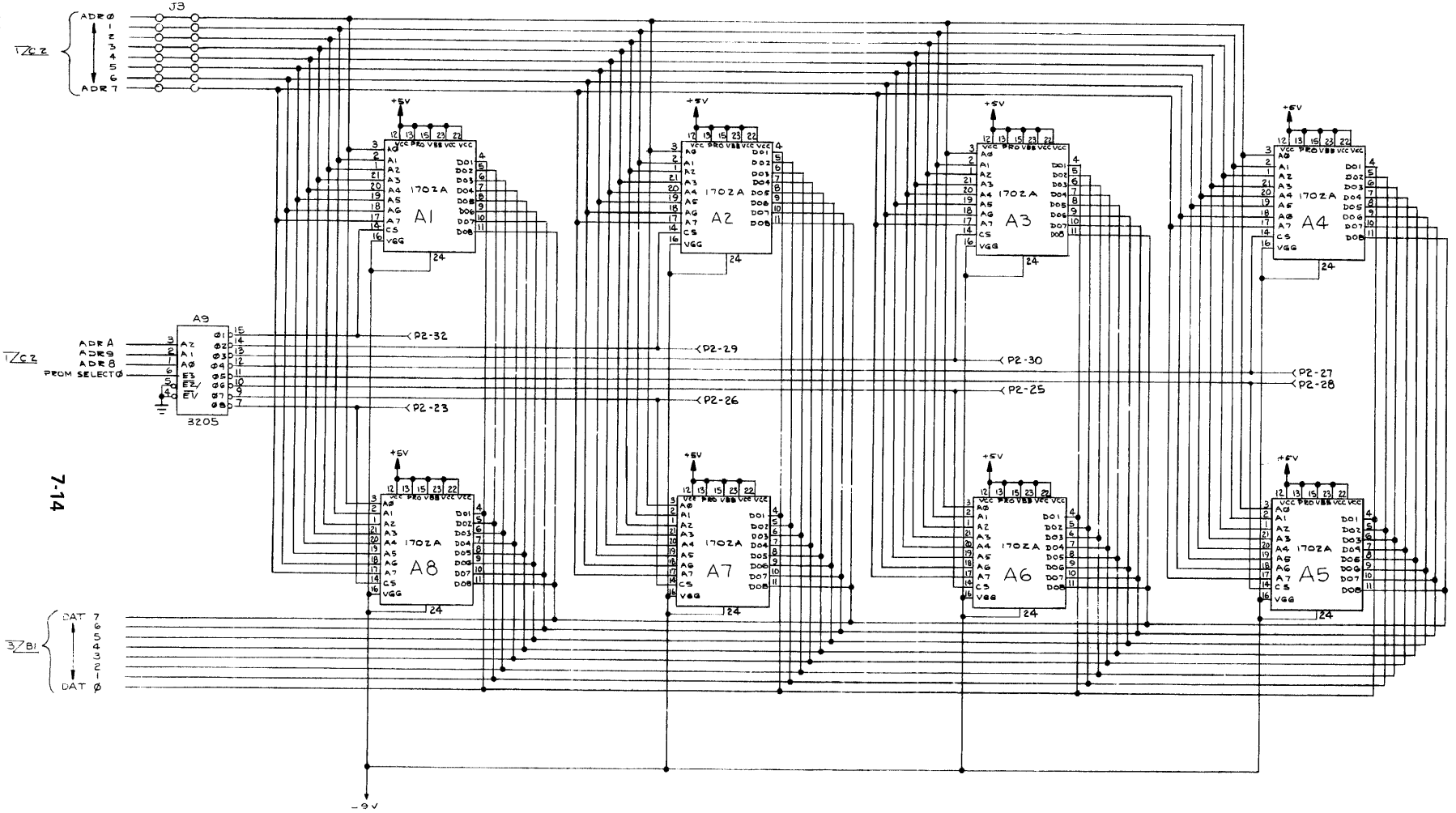


Figure 7-6. PROM Module Schematic (Sheet 2 of 5)

SCALE:	SIZE	DEPT	DRAWING NO.	REV
SHEET 2 OF 5	D	410	2000315	B

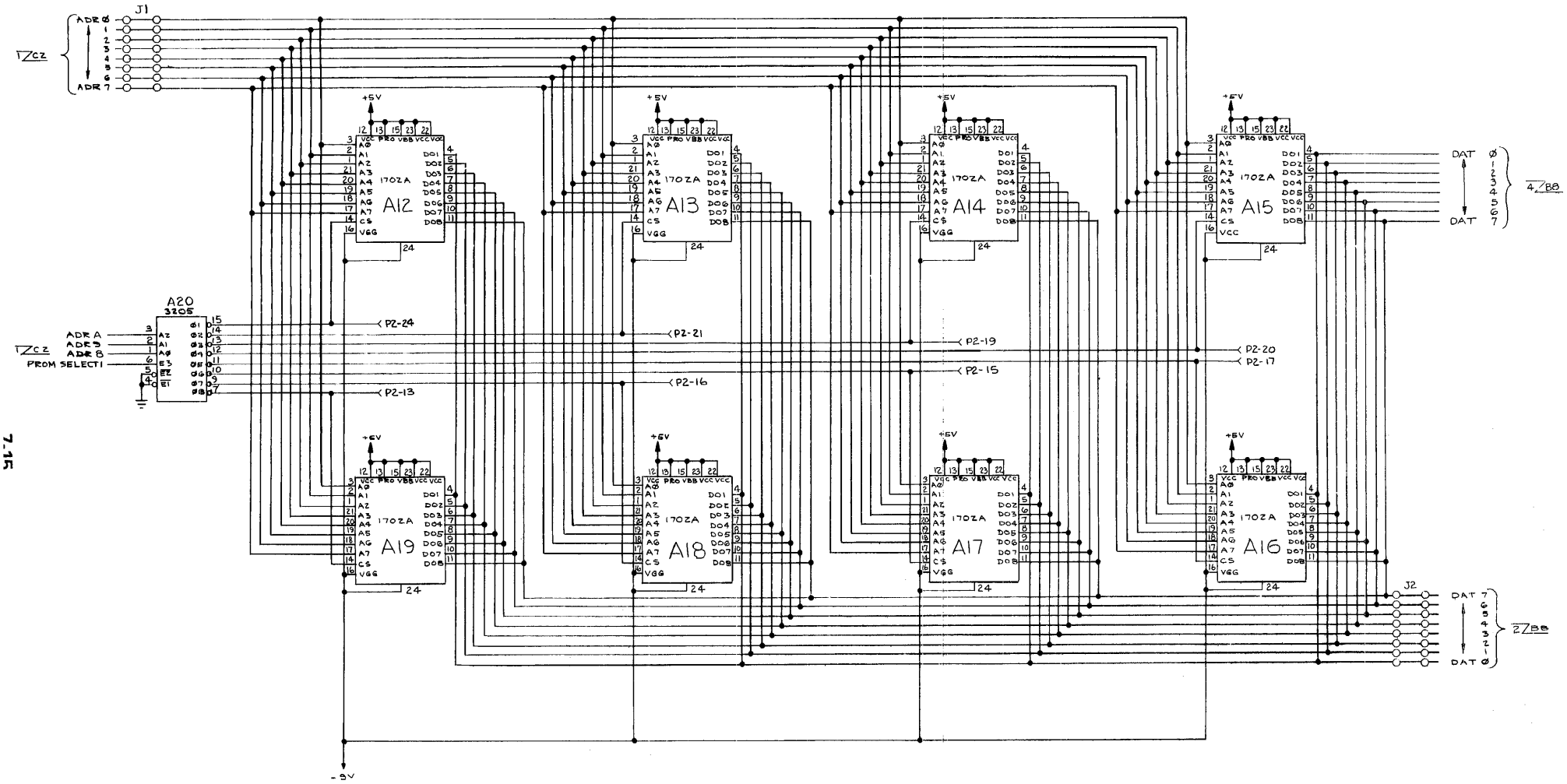


Figure 7-6. PROM Module Schematic (Sheet 3 of 5)

SCALE: —	SIZE	DEPT	DRAWING NO.	REV
SHEET 3 OF 5	D	410	2000315	B

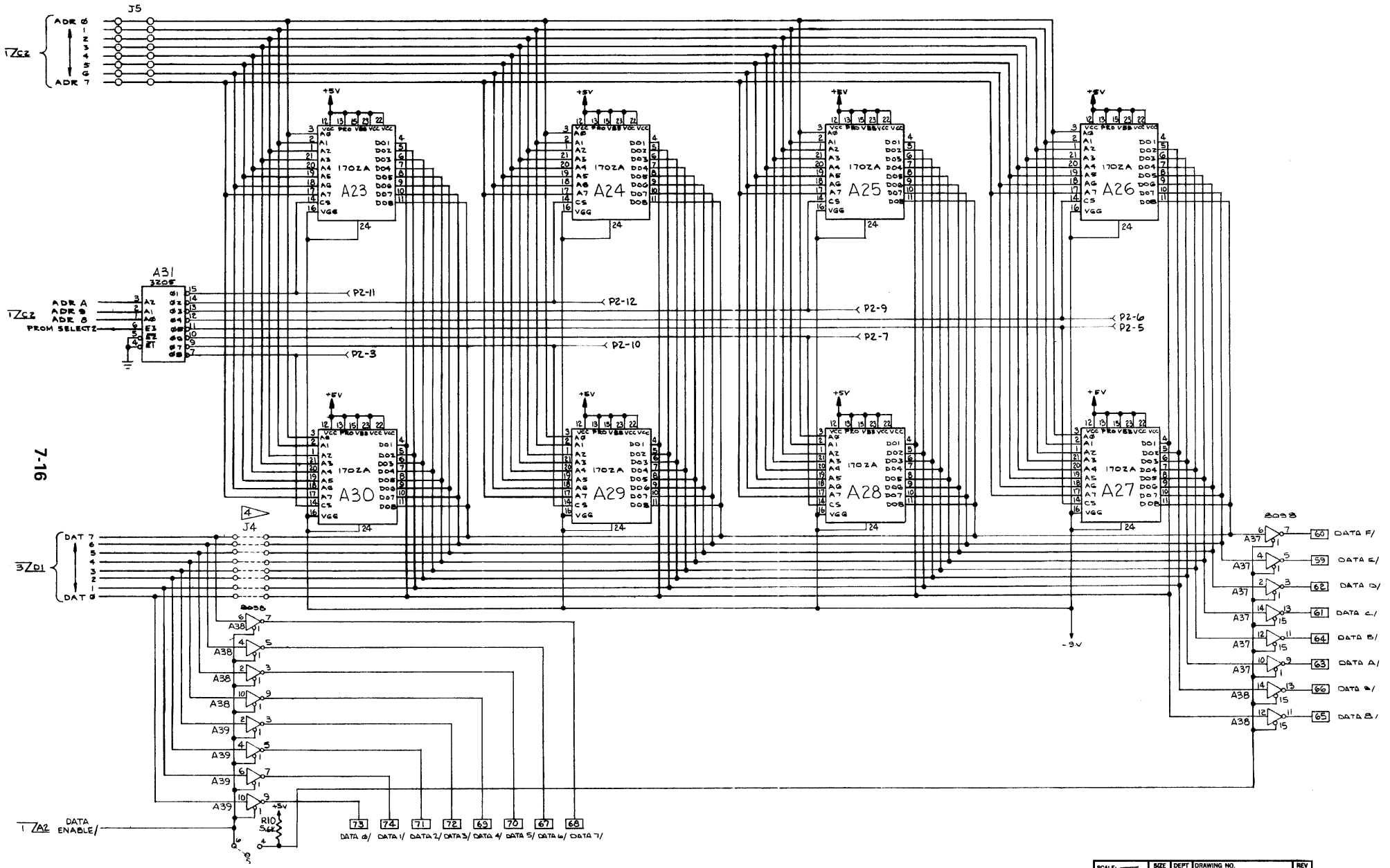


Figure 7-6. PROM Module Schematic (Sheet 4 of 5)

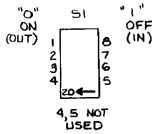
SCALE:	SIZE	DEPT	DRAWING NO.	REV
SHEET 4 OF 5	D	410	2000315	B.

ADDRESS SWITCH POSITION TABLE

ADDRESS		4K BANK $X_1 + X_2$			2K BANK $Y_1 + Y_2$			
		X_1 (S4)	X_2	LOCATION	Y_1 (S3)	Y_2	2K BANK MOD	LOCATION
ØØØ	7FF	0	8-9	A1 - A8	0	11-12	2-1	A23 - A30
8ØØ	FFF	0	↑	A12 - A19	0	↑	2-3	↑
1ØØØ	17FF	1	↑	A1 - A8	1	↑	2-1	↑
18ØØ	1FFF	1	↑	A12 - A19	1	↑	2-3	↑
2ØØØ	27FF	2	↑	A1 - A8	2	↑	2-1	↑
28ØØ	2FFF	2	↑	A12 - A19	2	↑	2-3	↑
3ØØØ	37FF	3	↑	A1 - A8	3	↑	2-1	↑
38ØØ	3FFF	3	↑	A12 - A19	3	↑	2-3	↑
4ØØØ	47FF	4	↑	A1 - A8	4	↑	2-1	↑
48ØØ	4FFF	4	↑	A12 - A19	4	↑	2-3	↑
5ØØØ	57FF	5	↑	A1 - A8	5	↑	2-1	↑
58ØØ	5FFF	5	↑	A12 - A19	5	↑	2-3	↑
6ØØØ	67FF	6	↑	A1 - A8	6	↑	2-1	↑
68ØØ	6FFF	6	↑	A12 - A19	6	↑	2-3	↑
7ØØØ	77FF	7	↓	A1 - A8	7	↓	2-1	↑
78ØØ	7FFF	7	8-9	A12 - A19	7	11-12	2-3	↑
8ØØØ	87FF	0	8-7	A1 - A8	0	11-10	2-1	↑
88ØØ	8FFF	0	↑	A12 - A19	0	↑	2-3	↑
9ØØØ	97FF	1	↑	A1 - A8	1	↑	2-1	↑
98ØØ	9FFF	1	↑	A12 - A19	1	↑	2-3	↑
AØØØ	A7FF	2	↑	A1 - A8	2	↑	2-1	↑
A8ØØ	AFFF	2	↑	A12 - A19	2	↑	2-3	↑
BØØØ	B7FF	3	↑	A1 - A8	3	↑	2-1	↑
B8ØØ	BFFF	3	↑	A12 - A19	3	↑	2-3	↑
CØØØ	C7FF	4	↑	A1 - A8	4	↑	2-1	↑
C8ØØ	CFFF	4	↑	A12 - A19	4	↑	2-3	↑
DØØØ	D7FF	5	↑	A1 - A8	5	↑	2-1	↑
D8ØØ	DFFF	5	↑	A12 - A19	5	↑	2-3	↑
EØØØ	E7FF	6	↑	A1 - A8	6	↑	2-1	↑
E8ØØ	EFFF	6	↑	A12 - A19	6	↑	2-3	↑
FØØØ	F7FF	7	↓	A1 - A8	7	↓	2-1	↓
F8ØØ	FFFF	7	8-7	A12 - A19	7	11-10	2-3	A23 - A30

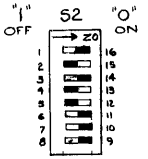
PROM RESIDENCY (S1)

BANK	LOCATION	SWITCH PIN NO.	1702A INSTALLATION	
			OUT	IN
2K	A23 - A30	1, 8	0	1
4K	A1 - A8	2, 7	0	1
	A12 - A19	3, 6	0	1



(S2) TIMING SWITCH SETTING TABLE

DEVICE	ACCESS TIME	1/16	2/15	3/14	4/13	5/12	6/11	7/10	8/9
1702A	1.0 US	0	1	0	1	1	0	0	0
1702A-S614	1.5 US	1	0	1	0	1	0	0	0
1702A-S314	1.7 US	0	0	1	0	1	0	0	0
1702A-S714	2.5 US	1	1	0	1	0	0	0	0



SETTING SHOWN IS FOR 1702A DEVICE

Figure 7-6. PROM Module Schematic (Sheet 5 of 5)

SCALE: —	SIZE	DEPT	DRAWING NO.	REV
SHEET 5 OF 5	D	4105	2000315	B

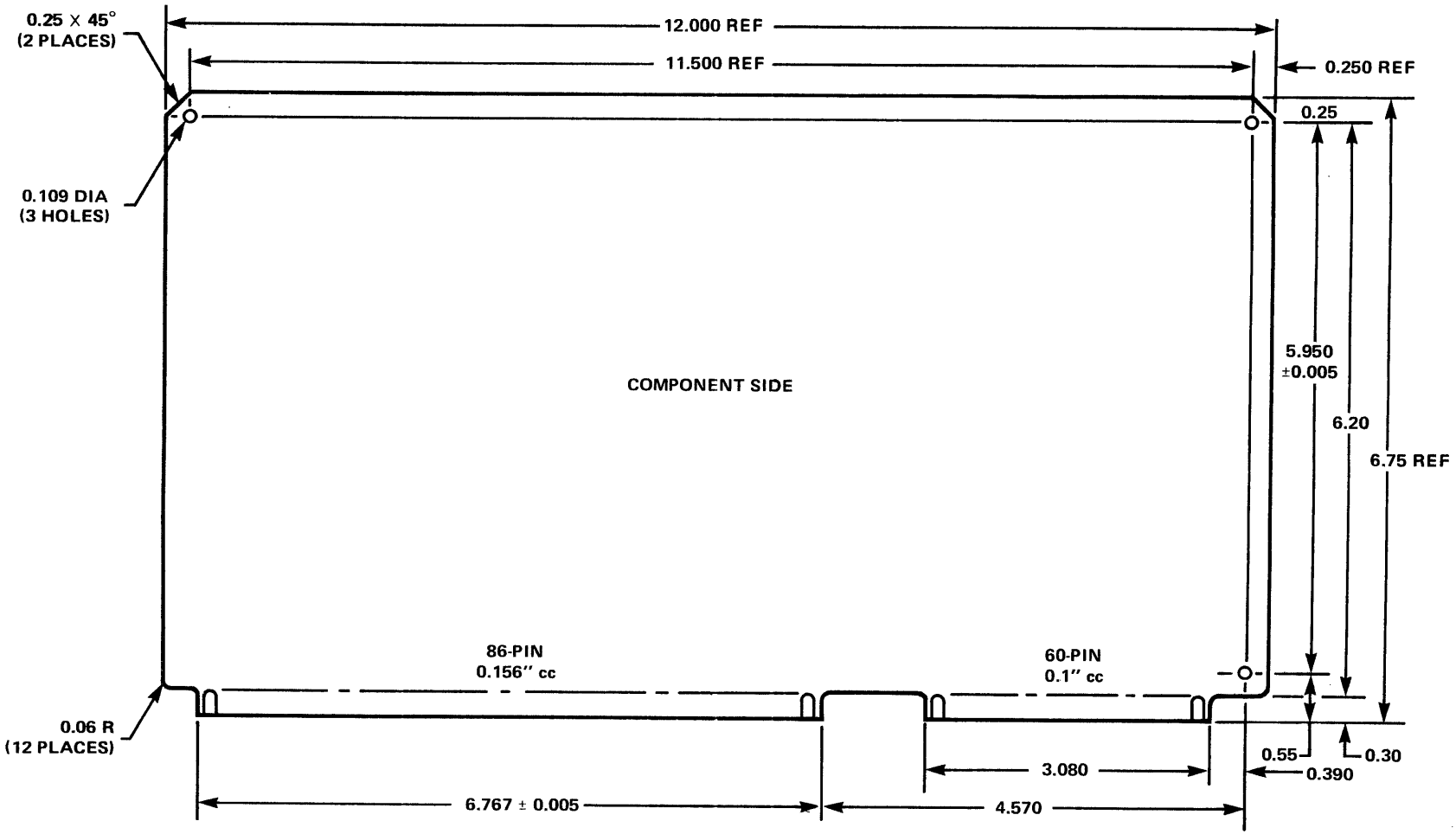


Figure 7-7. PROM Module Connectors

Signal Requirements

All data and control functions appearing at the module edge connectors are at TTL levels. Electrical characteristics of the signal inputs and outputs, as well as power inputs, are given in Section 7.4.

Signal descriptions and connector pin allocations are given in Section 7.3.2.

Address Assignments

The storage elements on the PROM Module are organized into a 4096 (4K) X 8-bit memory bank (16 PROM devices) and a 2048 (2K) X 8-bit memory bank (8 PROM devices).

The user must assign memory addresses to the 4K bank (if it is present) by:

- Setting the X1 switch (S4) to the appropriate position; and
- Joining the appropriate connections on the X2 jumper pad (7-8-9),

as defined in Table 7-1 (Section 7.2.2).

In addition, the PROM RESIDENT SELECT switch (S1) must be such that it reflects the PROM chip locations which are actually occupied, as described in Table 7-2 (Section 7.2.3).

Figure 7-8 illustrates the locations of the various switches and jumper pads on the PROM Module.

Access Timer Selection

There are several versions of the 8702A PROM device, with each version having a different access time. The PROM Module has been designed to operate with any one version. The user must, however, set the ACCESS TIMER SELECT switch (S2) on the PROM Module (see Figure 7-8) as defined in Table 7-3 (Section 7.2.3). The ACCESS TIMER SELECT switch dictates timing of the acknowledge signals, XACK/ and 8080/.

Byte Selection

The PROM Module can be used to store 8 or 16-bit words. The module can provide a maximum capability of 6,144 (6K) X 8-bit words (using 24 PROM's) or 2048 (2K) X 16-bit words (using 16 PROM's).

If 16-bit storage is required, the user must assign the same addresses to the 2048 locations in the 4K memory bank (8 PROMs) as are assigned to the 2K memory bank (also 8 PROMs) as explained in Section 7.2.2 (Table 7-1). In addition, jumper points 4-6 should be connected to provide a pathway via which the DATA ENABLE/ signal can enable the eight 8098 TRI STATE hex inverters that drive the high-order byte of the 16-bit word (from the 2K memory bank) on the data bus (DAT8/–DATF/). Because the high-order byte from the 2K memory bank is being driven on data lines DAT8/–DATF/, there is no need for maintaining a data path between the 2K memory bank and the low-order data lines DATØ/–DAT7/. The low-order byte from the 4K memory bank is driven on these low-order data lines. Consequently, the eight-pair jumper pad, J4, should be removed in 16-bit configurations. The DATA ENABLE/ and the J4 jumper connections are shown on sheet 4 of the module schematic, Figure 7-6.

7.3.2 PIN LISTS: PROM MODULE

The following section provides connector pin allocations for the PROM Module. The pins and their designated signal functions for the 86-pin connector (P1) are listed in Table 7-4. The same information for the 60-pin auxiliary connector (P2) is listed in Table 7-5.

7.4 OPERATING CHARACTERISTICS: PROM MODULE

This section provides detailed information concerning the AC and DC characteristics of the PROM Module.

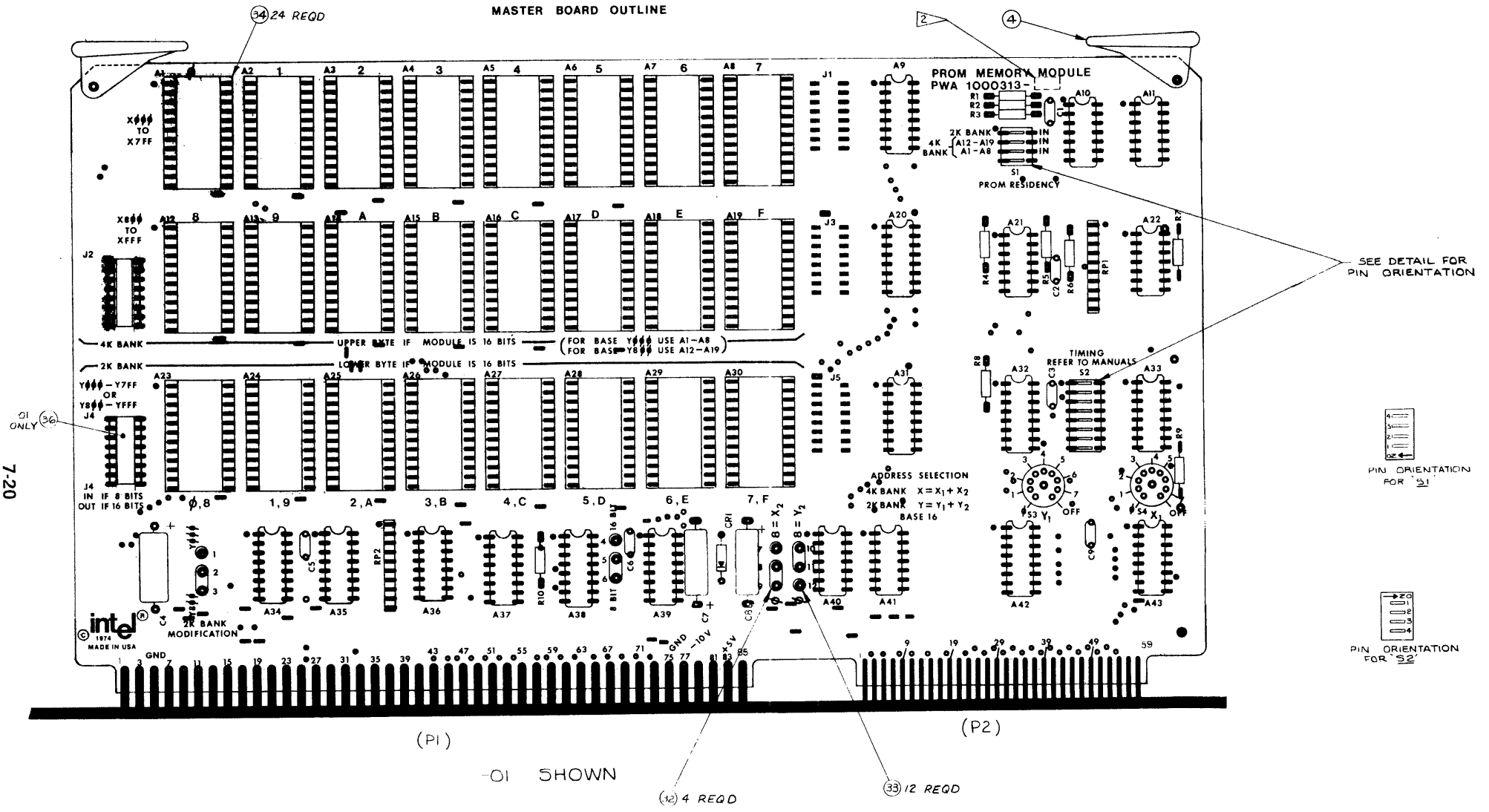
7.4.1 AC CHARACTERISTICS

The AC characteristics of the PROM Module are summarized in Figure 7-9.

7.4.2 DC CHARACTERISTICS

The DC characteristics of the PROM Module are summarized in Table 7-6. Power requirements are cited below:

	TYP	MAX
V_{CC} +5 VDC ±5%	1.54A	2.54A
V_{BB} –10 VDC ±5%	0.84A	1.45A



NOTE: UNLESS OTHERWISE SPECIFIED

1. ASSEMBLY NO. IS 1000313-XX
2. MARK DASH NO. IN POSITION SHOWN.

Figure 7-8. PROM Module Assembly Drawing

intel	3065 BOWERS AVE SANTA CLARA CALIF 95051
TITLE	PRINTED WIRING ASSEMBLY PROM MEMORY MODULE
D 410	1000313 C

Table 7-4

P1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	GND	Ground
2	GND	
3	+5 VDC	Power inputs
4	+5 VDC	
5	+5 VDC	
6	+5 VDC	
7		
8		
9		
10		
11	GND	Ground
12	GND	
13		
14		
15		
16		
17		
18		
19	MRDC/	Memory read command
20		
21		
22		
23	XACK/	Acknowledge transfer
24	INH1/	Inhibit RAM
25	AACK/	Advance acknowledge
26	INH2/	Inhibit ROM
27		
28		
29		
30		
31	CCLK/	Common clock (9,8304 MHz)
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43	ADRE/	Address bus

PIN	SIGNAL	FUNCTION
44	ADRF/	Address bus
45	ADRC/	
46	ADRD/	
47	ADRA/	
48	ADRB/	
49	ADR8/	
50	ADR9/	
51	ADR6/	
52	ADR7/	
53	ADR4/	
54	ADR5/	
55	ADR2/	
56	ADR3/	
57	ADRØ/	
58	ADR1/	
59	DATE/	Data bus
60	DARF/	
61	DATC/	
62	DATD/	
63	DATA/	
64	DATB/	
65	DAT8/	
66	DAT9/	
67	DAT6/	
68	DAT7/	
69	DAT4/	
70	DAT5/	
71	DAT2/	
72	DAT3/	
73	DATØ/	
74	DAT1/	
75	GND	Ground
76	GND	
77	-10 VDC	Power inputs
78	-10 VDC	
79		
80		
81	+5 VDC	Power inputs
82	+5 VDC	
83	+5 VDC	
84	+5 VDC	
85	GND	Ground
86	GND	

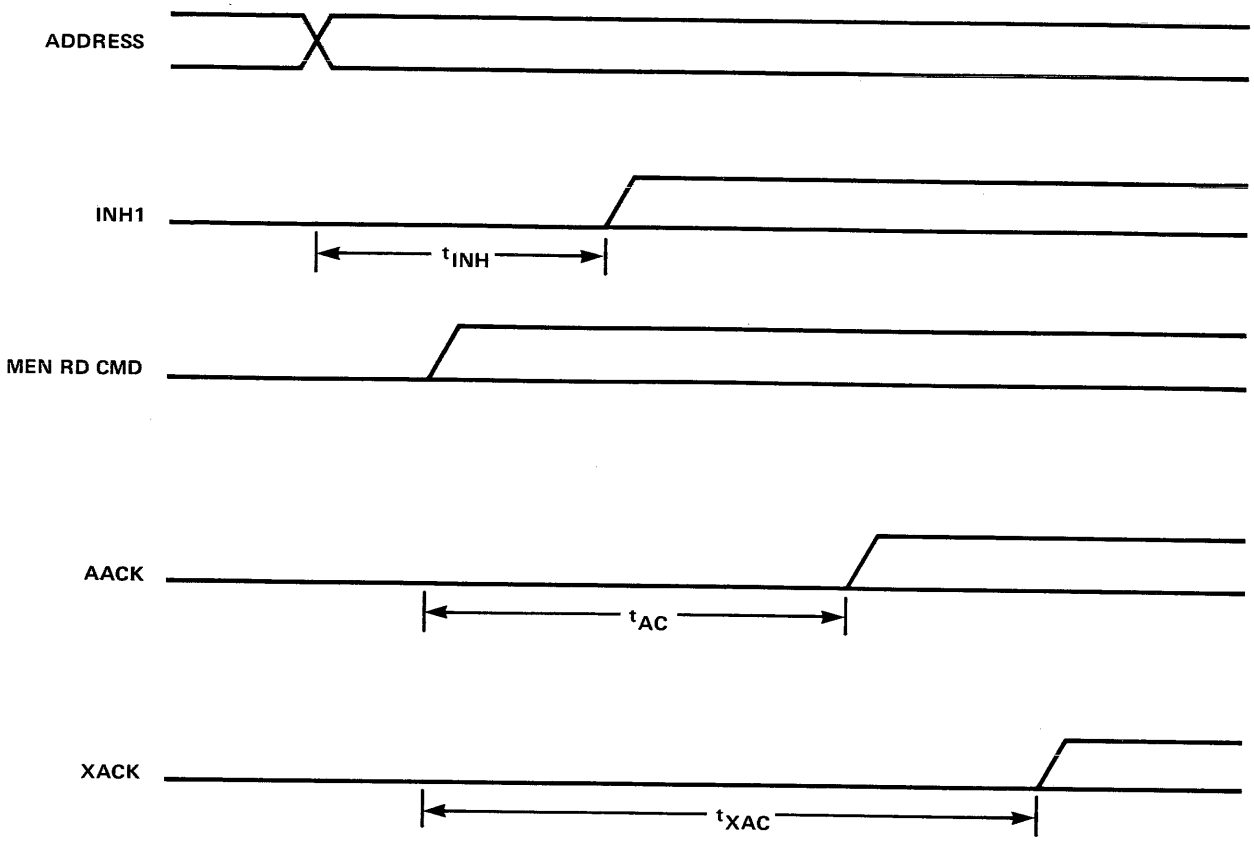
Table 7-5

P2 CONNECTOR PIN LIST (TEST POINTS)

PIN	SIGNAL*	FUNCTION
1	S4-10	
2		
3	CSA30	
4	S3-10	
5	CSA27	
6	CSA26	
7	CSA28	
8		
9	CSA25	
10	CSA29	
11	CSA23	
12	CSA24	
13	CSA19	
14		
15	CSA17	
16	CSA18	
17	CSA16	
18		
19	CSA14	
20	CSA15	TEST POINTS
21	CSA13	
22		
23	CSA23	
24	CSA12	
25	CSA6	
26	CSA7	
27	CSA4	
28	CSA5	
29	CSA2	
30	CSA3	

PIN	SIGNAL	FUNCTION
31	S1-1	
32	CSA1	
33	S1-3	
34	S1-2	
35	S3-8	
36	S2-13	
37	S3-7	
38	S3-6	
39	S3-5	
40	S3-4	
41	S3-2	
42	S3-4	
43	S2-15	
44	S3-1	
45	S2-14	
46	S2-16	
47	S2-11	
48	S2-12	
49	S2-9	
50	S2-10	
51	S4-2	TEST POINTS
52	S4-1	
53	S4-4	
54	S4-3	
55	S4-8	
56	S4-3	
57	S4-6	
58		
59	S4-7	
60		

*Designations refer to devices shown on the schematic. For example, pin 40 lists S3-4 which refers to position 4 on switch S3, shown on sheet 1 of the module schematic.



		8702A	8702A-S614	8702A-S314	8702A-S714
INH1	t_{INH}	102 ns	102 ns	102 ns	102 ns
AACK	t_{AC}	700 ns	1.2 µs	1.3 µs	2.3 µs
XACK	t_{XAL}	1.1 µs	1.6 µs	1.7 µs	2.8 µs

Figure 7-9. AC Characteristics of the PROM Module

Table 7-6

DC CHARACTERISTICS OF THE PROM MODULE

SIGNALS (DEVICE)	PARAMETERS	MIN.	MAX.	UNIT	TEST CONDITIONS
ADR0/–ADRF/ RD CMD/ (3404)	V_{IL}-Input low voltage	2.0	0.85	V	V_{CC} = 5.0V
	V_{IH}-Input high voltage			V	V_{CC} = 5.0V
	I_R-Input leakage current		10	μA	$V_{CC} = V_R$ = 5.25V
	I_F-Input load current		–0.25	mA	V_{CC} = 5.25, V_R = 0.45V
INH2/ (7408)	V_{IL}-Input low voltage	2.0	0.8	V	
	V_{IH}-Input high voltage			V	
	I_R-Input leakage current		1	mA	V_{CC} = 5.25V, V_R = 5.5V
	I_F-Input load current		–1.6	mA	V_{CC} = 5.25V
INH1/ (7403)	V_{OL}-Low level output voltage		0.4	V	V_{CC} = 4.75V, V_{IH} = 2V,
	V_{OH}-High level output voltage		*		I_{OL} = 16 mA
XACK/ AACK/ DATØ/–DATF/ (8093)	V_{OL}-Low level output voltage	2.4	0.4	V	V_{CC}= 4.5V, I_{OL} = 32 mA
	V_{OH}-High level output voltage			μA	V_{CC} = 4.5V, I_{OH} = –5.2 mA
	I_{LH}- Input current at high voltage		–40	μA	V_{CC} = 5.25V, High Z, V_R = 0.5, DIS = 2.0V
	I_{LL}-Input current at low voltage		40	μA	V_{CC} = 5.25V, High Z, V_0 = 2.4V
CCLK/ (74H04)	V_{IL}-Input low voltage	2		V	V_{CC} = 5.25V
	V_{IH}-Input high voltage		0.8	V	V_{CC} = 5.0V
	I_R-Input leakage current		1	mA	V_{CC} = 5.25V, V_I = 5.5V
	I_F-Input load current		–2.0	mA	V_{CC} = 5.25V, V_{IL} = 0.4V

*Open connector 10K P.U.

Chapter 8
DIRECT MEMORY ACCESS (DMA) MODULE

The DMA Module provides a direct memory access capability for the high-speed transfer of 8 or 16-bit data. Once a DMA operation is initiated by the central processor unit (CPU), the DMA Module controls the actual transfer of up to 65,536 words of data between memory and an external device, without any further intervention of the CPU required. The DMA Module can "steal" cycles by requesting control of the system bus for each word transferred; master control of the bus is granted on a priority basis. In addition, the CPU can, prior to the beginning of a transfer operation, invoke an override capability for the DMA Module. In this case, the DMA Module retains control of the bus until the entire block of data is transferred. After the entire transfer is completed, the CPU would, in response to a DMA interrupt, reset the override capability. This mode of operation allows for "burst" mode transfers to/from very high-speed peripherals.

While the data paths between the DMA Module and the external devices are only 8 bits wide, the bidirectional data bus between the module and the CPU can be either 8 or 16 bits wide. When transferring 16-bit data from memory to a device, the DMA Module disassembles the word and transfers the two bytes separately. Conversely, the module assembles two consecutive bytes from a device and sends the 16-bit word in parallel to memory.

The DMA Module includes provisions that allow it to be interrupt driven. In fact, the DMA interrupt request can be asserted on any one of eight interrupt priority levels. A DMA interrupt request can originate in the external device, in the DMA Module itself (upon completion of a transfer operation) or can be generated by the program being executed in the CPU. The CPU program can enable/disable interrupts or reset an existing interrupt request. The external device can also disable the presentation of interrupts to the CPU by the DMA Module.

In addition to providing a high-speed data path between memory and peripheral devices, the DMA Module includes five I/O ports that allow the CPU to directly address and access five devices (or groups of devices). The fifth port is associated with a 4-bit tag register. When this fifth I/O port is addressed, the contents of the tag register can be used to "steer" the input or output strobe to one of 16 additional devices; thus expanding the I/O capability of the DMA Module. These I/O ports between the CPU and the peripheral devices are usually used to initialize or test the device for a transfer operation, by sending control or address information (e.g., the sector address for a disk) to the device or by reading a status word from the device.

While the DMA Module has been designed specifically as an option for use in INTELLEC MDS Systems, the module is flexible and powerful enough to be used in many different 8 or 16-bit computer systems that require a direct memory access capability. Consequently, the DMA Module, like all other INTELLEC Modules, is independently available on an OEM basis.

The module is implemented on a single 12-in. X 6.75-in. printed circuit board. The module requires only +5 VDC power.

8.1 FUNCTIONAL ORGANIZATION OF THE DMA MODULE

The DMA Module can be viewed, for descriptive purposes, as consisting of 11 functional blocks:

(1) Address decoding logic
(2) XACK/ generation logic
(3) Control register
(4) Length register
(5) Memory address register
(6) Tag register
(7) Status logic
(8) Bus in/out logic
(9) DMA interrupt logic
(10) DMA transfer control logic
(11) Bus interface logic

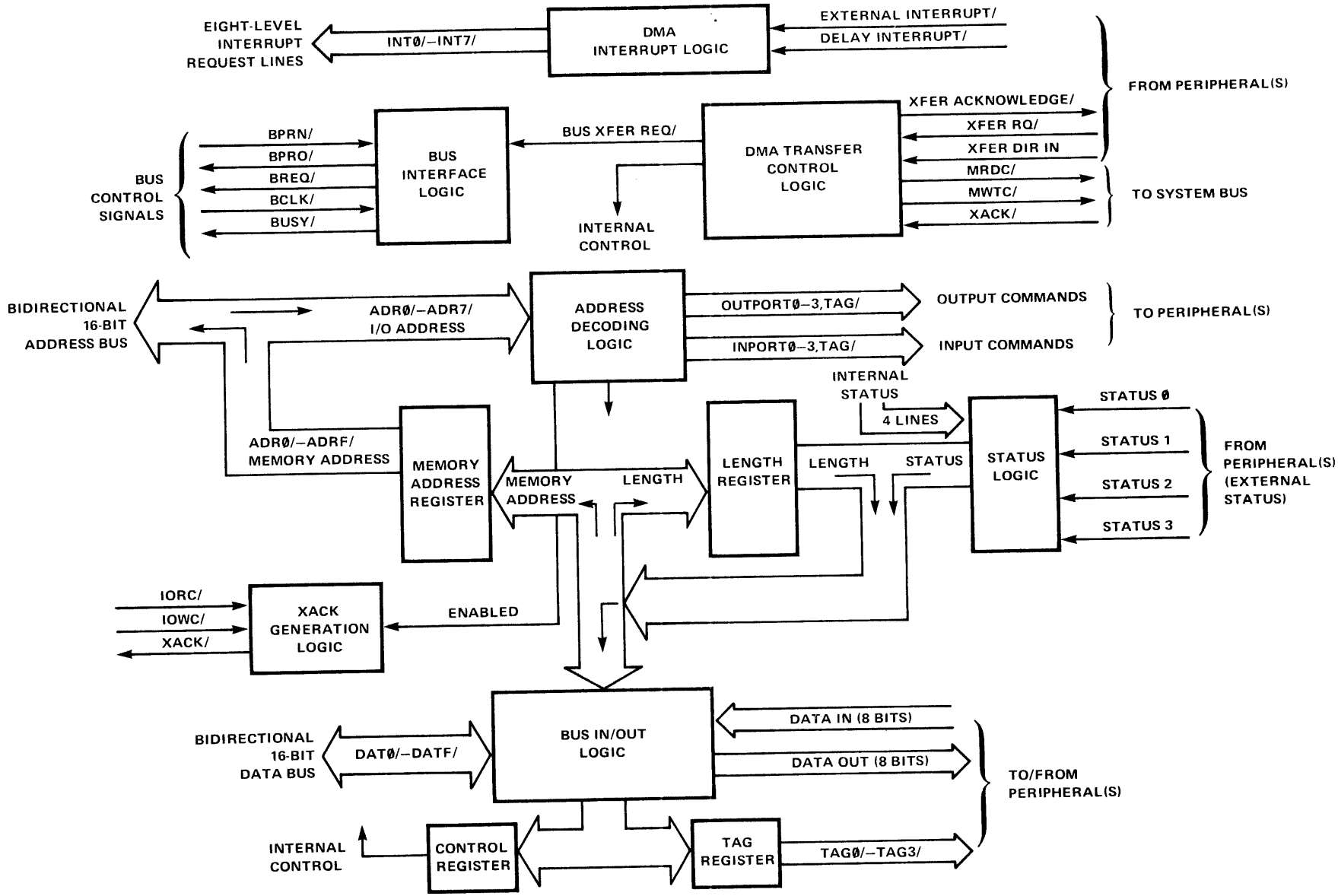


Figure 8-1. DMA Module Functional Block Diagram

as illustrated in Figure 8-1.

Before beginning a DMA transfer operation, the DMA Module must be initialized by the CPU. Only after the proper parameters have been loaded into the module can the actual block transfer begin. Once initialized by the CPU, however, the DMA Module can transfer up to 65,536 words of data (8 or 16-bit) between memory and an external peripheral device without any further intervention required of the CPU.

The execution of a direct memory access (DMA) operation requires considerable interaction between the 11 functional blocks that we listed. Consequently, before defining the specific responsibilities of each functional block, we will step through the general sequence of events in a DMA operation.

Sequence of Events

A typical DMA operation is initiated and executed as follows:

(1) The CPU reads the status of the DMA Module and the peripheral device to determine that both are available for use. The CPU reads status by executing an I/O read (input) instruction addressed to port "BASE+6", where BASE is the switch-selectable base address of the DMA Module. (In a multi-processor system, the CPU should use its bus override feature during the busy test/set sequence to prevent detrimental interaction with the DMA Module by other CPUs.)

(2) The CPU loads a control word into the DMA Module's control register. This sets the busy indicator and specifies the type of operation to be performed. The control register is accessed by executing an I/O write (output) instruction addressed to port "BASE+A_{16}".

(3) The CPU outputs a 16-bit value (two 8-bit bytes), specifying the number of words to be transferred. This value is loaded into the 16-bit length register. The least significant byte is loaded by executing an I/O write (output) instruction addressed to port "BASE+6". The most significant byte is located by executing an output instruction to port "BASE+D".

(4) The CPU outputs a 16-bit memory address, specifying the first memory location to be accessed. The least significant byte is loaded into the memory address register by executing an I/O write (output) instruction to port "BASE+E". The most significant byte is loaded by executing an output instruction to port "BASE+F".

(5) The CPU must also provide the peripheral device with any parameters it may need for the operation (e.g., the sector number for a disk). When the CPU executes an output instruction to address "BASE+port#" (where port# equals 0,1,2,3 or TAG), the DMA Module routes the byte from the CPU to the peripheral device and generates an output pulse that strobes the byte into the device.

(6) The last step in the initialization portion of the operation is the issuance of a "go" command to the peripheral device. Normally, the CPU issues a "go" command by outputting a command byte to the appropriate port as described in Step 5, but in some configurations a bit in the tag register could be used to notify the device. The CPU can load the tag register by executing output instructions to port "BASE+B_{16}".

(7) Each time that the external device requires transfer of a data word, it issues a transfer request (XFER RQ/) to the DMA Module's DMA transfer control logic. The DMA Module's bus interface logic, in turn, requests use of the system bus. After the DMA Module gains control of the system bus, the DMA transfer control logic issues a memory read (MRDC/) or write (MWTC/) command. After the data word is transferred to/from the peripheral device (via the DMA Module's bus in/out logic), the DMA transfer control logic issues a transfer acknowledge signal (XFER ACKNOWLEDGE) to the peripheral device. The DMA Module decrements the word count in its length register and increments the value in its memory address register. This

sequence continues until the word count equals zero.

(8) When the word count in the DMA Module's length register is decremented to zero, the interrupt latch in the DMA Module is set. If the peripheral is asserting its delay interrupt signal (DELAY INT/), the interrupt request is not passed on to the CPU until the delay signal goes false. If the enable interrupt bit in the DMA Module's control register is set, the interrupt request can be asserted on any one of the eight interrupt levels (the level is switch selected). If DMA interrupts are not enabled, the CPU must periodically interrogate DMA status to determine when the transfer has been completed.

(9) When the CPU determines that the transfer is complete, it calls a service routine. At the end of the service routine, the CPU issues a reset interrupt command to the DMA Module by executing an output instruction to port "BASE+9". The reset interrupt command resets the DMA Module's busy latch and the interrupt latch. The reset interrupt command will also reset the set interrupt latch if the CPU had initiated an interrupt request in the DMA Module. The DMA Module is now ready for the next operation.

Functional Block Descriptions

As we saw in the above sequence of events, the CPU accesses the various blocks within the DMA Module by executing I/O instructions directed to a switch-selectable block of 16 dedicated port addresses. The *address decoding logic* receives the port address output by the CPU on the bidirectional address bus and generates the appropriate internal control signal for the DMA Module, or generates the proper input/output strobe for the peripheral device. The CPU only accesses the DMA Module to initialize it prior to a transfer operation or to reset it after a transfer operation has been completed. The DMA Module, itself, controls the actual transfer of data between memory and an external peripheral device.

Whenever the CPU accesses an I/O port, it waits for the port to return a transfer acknowledge signal (XACK/) before completing the execution of the I/O instruction. The *XACK/ generation logic*, as its name implies, is responsible for returning the XACK/ signal, whenever the CPU executes an I/O instruction to a port address within the range defined by the DMA Module base address. The actual timing for the generation of XACK/ is jumper-selectable. In addition to generating XACK/ in response to I/O instructions from the CPU, the XACK/ generation logic accepts XACK/ signals generated by memory during DMA data transfers and passes this acknowledgement on to the DMA transfer control logic.

The *control register* is loaded by the CPU, prior to the beginning of a DMA operation. The contents of the control register specify the busy status of the DMA Module, the type of operation to be performed (transfer or non-transfer), the transfer direction (to or from memory), the word size (8 or 16 bits), the interrupt condition (enabled or disabled), and the means by which the DMA Module is using the system bus (contention basis or override basis).

The 16-bit *length register* is loaded by the CPU, prior to a transfer operation, with the word count value specifying the total number of words to be transferred. The word count is decremented by one after each word is transferred. The transfer stops when the word count equals zero.

The 16-bit *memory address register* is loaded by the CPU, prior to a transfer operation, with the address of the first memory location to be accessed. The address is incremented by one for each word transferred. The current 16-bit address is gated onto the bidirectional system address bus and sent to memory during each transfer sequence.

The *tag register* is a general-purpose 4-bit register. The contents of the tag register can be made available to all of the external peripheral devices being controlled by the DMA Module. The tag register can be used in a number of ways. The four tag lines can be used as control lines to the devices (e.g., as the "go" command line to each of four devices). The tag register might be used to expand the maximum number of DMA peripherals by 16. In this case, the tag register would store the select code for one of 16 additional devices. Only that device which recognized its 4-bit select code on the tag

lines would respond to command bytes output on the data out bus with the OUTPUT TAG/ strobe.

The *status logic* groups together four internal status bits from the DMA Module (SET INT/, MEM WRT/, INTERRUPT STATUS/, and DMA BUSY/) and four status bits from the external peripheral device, and multiplexes these 8 bits onto the system data bus when directed to do so by the CPU (i.e., when the CPU executes the IN PORT instruction, where PORT = BASE+6).

The *bus in/out logic* routes all data flow to, from or through the DMA Module to its intended destination. Control or address information that is output on the system data bus by the CPU is buffered in the bus in/out logic and directed to registers within the DMA Module (e.g., the length register), or routed out to the peripheral devices via the data out bus (e.g., a "go" command word). Data output by memory during a DMA transfer sequence is received from the bidirectional system data bus, buffered, then routed onto the data out bus and out to an external device. Data input by a peripheral device during a DMA transfer sequence is received from the data in bus, buffered, then routed onto the bidirectional system data bus which carries the data to memory. Status, address or word count information from the DMA Module or from an external device is directed through the bus in/out logic and onto the system data bus which carries it to the CPU.

The *DMA interrupt logic* accepts interrupt requests generated by the CPU program, an external device, or by the DMA Module at the end of a DMA transfer operation (i.e., when the length register contents equal zero). The DMA interrupt logic will only respond (i.e., set the interrupt latch) to an interrupt request originating in the CPU program if the DMA Module is not busy. If the contents of the DMA Module's control register indicate that interrupts are enabled, an interrupt request (regardless of its source) can be asserted on any one of the eight interrupt levels (INTØ/–INT7/). An interrupt level is selected by setting a rotary switch in the DMA interrupt logic to the desired position. If the DELAY INT/ line from the external devices is true however, the interrupt request to the CPU will be delayed until DELAY INT/ goes false.

The *DMA transfer control logic* is responsible for the "handshaking" exchanges with the peripheral device and memory during all DMA operations. The DMA transfer control logic accepts transfer request (XFER RQ/) and transfer direction (XFER DIR IN) signals from the external device, generates the read (MRDC/) or write (MWTC/) command for memory, and uses the memory's acknowledgement (XACK/) to generate a transfer acknowledge signal (XFER ACKNOWLEDGE/) for the external device. The DMA transfer control logic also notifies the bus interface logic when it needs the system bus for a data transfer (i.e., BUS XFER RQ/ goes true).

The *bus interface logic* allows the DMA Module to operate as a bus master; that is, to obtain control of the system bus at the exclusion of all other master modules, including the primary CPU. When the DMA Module requires the bus for a DMA data transfer (i.e., when BUS XFER RQ/ is true), the bus interface logic issues the bus request signal (BREQ/). When the bus priority in signal (BPRN/) indicates that no higher priority module is requesting the bus, the bus interface logic sets its bus busy latch (BUSY/) and informs the other logic on the DMA Module that it has been selected. The DMA Module now has control of the bus for one data word transfer sequence. If the DMA Module is to require uninterrupted use of the system bus (e.g., burst mode transfers), an override capability can be extended to the module by the CPU. Override is enabled by setting a bit in the DMA Module's control register. Operation of the bus interface logic is referred to the bus clock signal (BCLK/), which must be supplied by another module in the system.

8.2 DMA MODULE: THEORY OF OPERATION

The following sub-sections provide a complete description of the theory of operation for each of the functional units on the DMA Module.

The DMA Module accepts/transmits signals, data and power through three different PC edge connectors:

- J1 Peripheral connector (to/from I/O peripherals)
- P1 Bus connector (to/from the system bus)
- P2 Auxiliary connector (to/from the auxiliary bus)

To avoid any ambiguity when referring to connector pins in subsequent paragraphs, we will always list the connector as well as the pin whenever such references are required; for example, P1-14 refers to pin 14 on connector P1. Pin lists for the three connectors are provided in Section 8.3.2.

The schematic (3 sheets) for the DMA Module is provided in Figure 8-9, located in Section 8.2.12.

8.2.1 ADDRESS DECODING LOGIC

The address decoding logic consists of four 3205 three-to-eight decoders, a nine-position rotary switch, a jumper plug and assorted gating and buffer circuits, as shown on sheet 2 of the module schematic, Figure 8-9.

The CPU sends control or address information to, or receives status information from, the DMA Module or one of the attached peripheral devices by executing an I/O instruction directed to a dedicated 8-bit port address. The address decoding logic examines address lines ADRØ/–ADR7/ (pins P1-51 through P1-58) and generates the appropriate control or strobe signal.

Address line ADR7/ is applied to one pole of the X2 jumper plug in an active-low state, and is also inverted and applied to another pole of X2 in an active-high state. Address lines ADR4/, ADR5/ and ADR6/ are applied to the address inputs of one 3205 decoder. The three address inputs cause a low level to appear on one of the eight decoder outputs. Each of these outputs are tied to one position of the X1 rotary switch (S2); position 9 is considered off. The outputs from the X1 switch and the X2 jumper plug feed a 7427 negative-input AND gate. The output of the 7427 section (A64-12), in conjunction with address line ADR3/ and the I/O write (IOWC/) or I/O read (IORC/) command, enables the other three 3205 decoders in the address decoding logic. The range of port address that will actually activate the 7427 gate is dependent on the base address for the DMA Module (address bits 4–7) as determined by the positioning of the X1 switch and the X2 jumper. Table 8-1 correlates all of the possible switch and jumper positioning combinations with the base address that they define.

The three least significant address lines (ADRØ/–ADR2/) are applied to the address inputs on three other 3205 decoders. As we mentioned above, address line ADR3/ (active-low) or its complement, ADR3 (active-high), feeds one of the enable inputs on the three decoders. The I/O write command (IOWC/) feeds the other enable input on the first two decoders (output ports), while the I/O read command (IORC/) feeds an enable input on the third decoder (input ports).

The first decoder produces the five output strobes, OUTPORTØ/–OUTPORT3/ and OUTPORT TAG/. The five most significant decoder outputs are buffered, inverted and gated with OUTP PULSE to produce the OUTPORT strobes. OUTP PULSE is generated in the XACK/ generation logic. Timing for the OUTPORT strobes is dependent on the setting of the time base jumper pad in the XACK/ generation logic; that is, an OUTPORT strobe will occur 100, 200, 400, 800 or 1600 ns after the I/O write command (IOWC/) is received, and will remain true for 200, 400, 800, 1600, or 3200 ns, respectively, according to the time base setting (see Section 8.2.2, Figure 8-2).

The eight outputs from the second decoder constitute eight internal control signals for the DMA Module, as listed in Table 8-2.

The eight outputs from the third decoder form the five input strobes to the external devices, INPORTØ/–INPORT3/ and INPORT TAG/, as well as the read DMA status (RD DMAC STAT/) and read length register (RD LEN MSBY/ and RD LEN LSBY/) internal command signals. RD LEN MSBY/ and RD LEN LSBY/ (read most and least significant bytes) are NORed together to form INPUT EN1/. INPUT EN1/, in turn, is NANDed with IORC, ADR3/, and the enable signal (ENABLED) from the 7427 base address gate (A64-12) to form the INPUT EN2/ signal. INPUT EN1/ allows the contents of the length register to be read, and INPUT EN2/ allows the DMA status word to be read (INPUT EN1/ and INPUT EN2/ enable multiplexers shown on sheet 3 of the module schematic).

Table 8-2 lists the internal control signals and external strobes, with the dedicated port addresses that select each one.

Table 8-1

DMA MODULE BASE ADDRESS SELECTION

BASE ADDRESS (HEX)	PORT ADDRESS RANGE (HEX)	X1 (S2 SWITCH SETTING)*	X2 (JUMPER 28-29-30)
00	00–0F	1	29–30
10	10–1F	2	
20	20–2F	3	
30	30–3F	4	
40	40–4F	5	
50	50–5F	6	
60	60–6F	7	
70	70–7F	8	29–30
80	80–8F	1	28–29
90	90–9F	2	
A0	A0–AF	3	
B0	B0–BF	4	
C0	C0–CF	5	
D0	D0–DF	6	
E0	E0–EF	7	
F0	F0–FF	8	28–29

*S2 Switch position 9 is OFF.

8.2.2 XACK/ GENERATION LOGIC

The XACK/ generation logic consists of two 74LS193 counters, a five-pair time base jumper pad, a 74S74 flip-flop and various gating circuits, as shown on sheet 1 of the module schematic, Figure 8-9.

The XACK/ generation logic is responsible for acknowledging, at the proper time, all I/O read (IORC/) and I/O write (IOWC/) commands directed to the DMA Module. The 9.8304 MHz common clock pulse (CCLK/) can be applied to the down-count input of the first of two 74LS193 counters or can feed the up-count input of the second counter directly (jumper pair 25-26 must be connected). If jumper pair 26-27 is not connected, the first counter divides the CCLK/ pulse by two (200 ns), four (400 ns), eight (800 ns) and 16 (1600 ns), at its QA, QB, QC and QD outputs, respectively. Each counter output is tied to one-half of a jumper pair. The particular jumper pair which is connected determines the frequency of the up-count input to the second counter. This counter is pre-loaded to 12_{10} (1100_2). The occurrence of IORC/ or IOWC/ causes the LD input to the counter to go false, allowing the count sequence to begin. Two counts later, the QB output from the counter goes high.

If IORC is true (i.e., if it is an I/O read operation), the D-input to a 74S74 latch goes high when QB goes high. On the next count pulse, the latch is clocked set and XACK/ goes true. The occurrence of XACK/ disables all further counting. XACK/ is available at pin P1-23 and remains true until IORC/ goes false.

If an I/O write operation is in progress, instead of a read, the D-input to the XACK/ latch does not go true until the QD counter output goes true; that is, eight count pulses after IOWC/ appeared. As before, the XACK/ is clocked set on the leading edge of the next count pulse. XACK/ goes true and

Table 8-2

I/O PORT ADDRESS DECODING

PORT ADDRESS (HEX)	I/O COMMAND	DECODER LOCATION	SIGNAL (PIN)	DESCRIPTION
BASE+0	IOWC/ (output)	A61	OUTPORT 0/ (J2-56)	Output strobes to external devices
BASE+1			OUTPORT 1/ (J2-54)	
BASE+2			OUTPORT 2/ (J2-52)	
BASE+3			OUTPORT 3/ (J2-50)	
BASE+4			OUTPORT TAG/ (J2-10)	
BASE+5				Not Used
BASE+6		A61		
BASE+7		A48		
BASE+8			SET INT/	Set the "SET INT" latch (also sets the INT latch if DMA not busy
BASE+9			RESET INTERRUPT/	Clears interrupt, DMA busy and SET INT latches
BASE+A			LOAD CNTRL REGISTER/	Load control register
BASE+B			LOAD TAG REGISTER/	Load tag register
BASE+C			OUTP LEN LSBY/	Load least significant byte of length register
BASE+D			OUTP LEN MSBY/	Load most significant byte of length register
BASE+E			OUTP MA LSBY/	Load LSBY of memory address register
BASE+F	IOWC/ (Output)	A48	OUTP MA MSBY/	Load MSBY of memory address register
BASE+0	IORC/ (input)	A49	INPORT O/ (J2-16)	Input strobes to external devices
BASE+1			INPORT 1/ (J2-18)	
BASE+2			INPORT 2/ (J2-20)	
BASE+3			INPORT 3/ (J2-22)	
BASE+4			RD LEN LSBY/	Read LSBY of length register
BASE+5			RD LEN MSBY/	Read MSBY of length register
BASE+6		A49	RD DMAC STAT/	Read DMA status word
BASE+7			INPORT RAG/ (J2-12)	Input strobe to external devices
BASE+8				Not Used
BASE+9				
BASE+A				
BASE+B				
BASE+C				
BASE+D				
BASE+E				
BASE+F	IORC/ (input)			

disables any further counting. XACK/ is available at pin P1-23 and remains true until IOWC/ goes false.

Figure 8-2 illustrates XACK/ timing for each of the five time base jumper pairs, during both I/O read and write cycles.

When the DMA Module, as bus master, is transferring a data word to or from memory, the memory module returns an XACK/ signal in response to the memory read (MRDC/) or write (MWTC/) command generated by the DMA Module. This XACK/ signal is received at pin P1-23 and made available to the DMA transfer control logic (see Section 8.2.10).

8.2.3 CONTROL REGISTER

The control register is a 3404 six-bit, high-speed latch, shown on sheet 2 of the module schematic, Figure 8-9.

The control register is normally loaded prior to the beginning of a DMA operation. When the CPU executes an I/O write instruction to port "BASE+ A_{16}", the DMA Module's address decoding logic generates the LOAD CNTRL REGISTER/ signal which is gated through to the write enable inputs on the 3404 six-bit latch. Data bits 0–3 from the CPU are buffered and inverted in the bus in/out logic and applied to four of the control register inverting data inputs. Data bits 4 and 5 are applied directly to the two remaining 3404 inverting data inputs. Bit definitions for the control register are as follows:

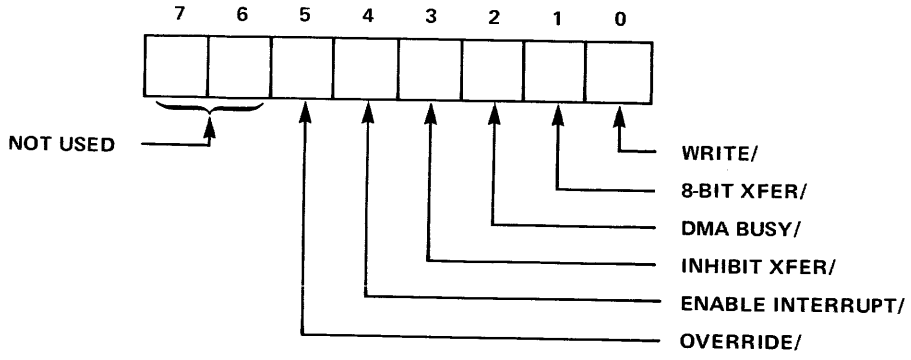
When the WRITE/ signal is true (low), it indicates that data is to be transferred from an external device to memory (WRITE MODE). When WRITE/ is high, it indicates that data is to be transferred from memory (READ MODE) to an external device.

When the 8 BIT XFER/ signal is true (low), it indicates that 8-bit data words are to be transferred. When 8 BIT XFER/ is high, it indicates that 16-bit data words are to be transferred.

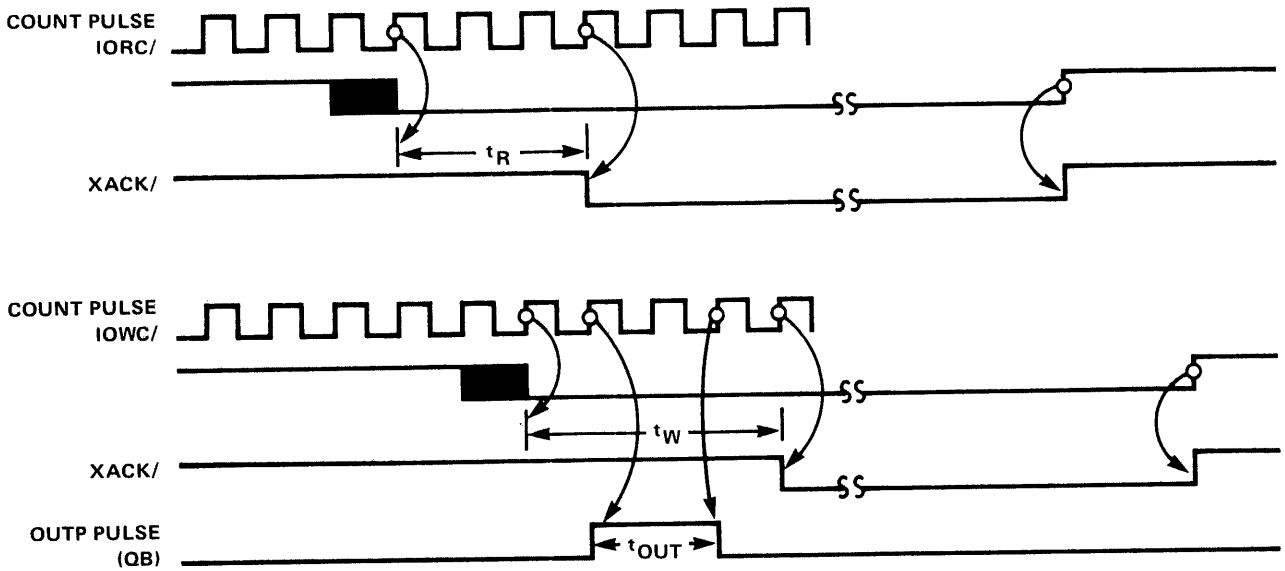
When DMA BUSY/ is true (low), it specifies that the DMA Module is engaged in an operation. The latch output actually feeds a 7474 flip-flop that is clocked by LOAD CNTRL REGISTER/ and preset by reset interrupt command (SYSINTRST/). The Q output of this 7474 section is labeled DMA BUSY/ (active-low), while the Q output is labeled DMA BUSY (active-high). Thus, the DMA Module is considered busy when this 7474 section is in the reset state.

When INHIBIT XFER/ is true (low), the DMA Module prevents data transfer from actually occurring. The module will, however, acknowledge all transfer requests (XFER RQ/) from the external device. When INHIBIT XFER/ is false, data will be transferred.

When ENABLE INTERRUPT is true (high), the DMA Module will issue an interrupt request to the CPU whenever it completes a DMA operation, or whenever it receives an interrupt command from the external device or from the CPU program (see Section 8.2.9). When ENABLE INTERRUPT is false, the interrupt latch will be set as usual, but no



NOTE: THE SOFTWARE DEFINITIONS OF ALL BITS ARE POSITIVE TRUE.



TIME BASE JUMPER PAIR CONNECTED	COUNT PULSE PERIOD (ns)	t_R (I/O READ)	t_W (I/O WRITE)	t_{OUT} (I/O WRITE)
25–26	100	200 ns	800 ns	200 ns
3–4	200	400 ns	1.6 μs	400 ns
1–2	400	800 ns	3.2 μs	800 ns
5–6	800	1.6 μs	6.4 μs	1.6 μs
7–8	1600	3.2 μs	12.8 μs	3.2 μs

Figure 8-2. XACK/ Generation Timing (Nominal)

interrupt request will be issued to the CPU. The status of the interrupt latch can be read by the CPU program, however.

Normally, the DMA Module requests use of the system bus for each word transferred. If, however, uninterrupted use of the bus is required, the CPU can set the override bit in the control register. If the override bit is set and the interrupt latch is not set, the OVERIDE/ signal will be true (low). When OVERIDE/ is true, the bus interface logic is prevented from relinquishing control of the system bus (see Section 8.2.11).

8.2.4 LENGTH REGISTER

The length register consists of four 74LS193 4-bit counters, as shown on sheet 3 of the module schematic, Figure 8-9. Prior to the beginning of a DMA transfer operation, the length register is loaded with the initial 16-bit word count (i.e., the total number of data words to be transferred).

The least significant byte of this initial word count value is output onto the system data bus when the CPU executes an I/O write instruction directed to port "BASE+C". The address decoding logic generates the OUTP LEN LSBY/ signal which is applied to the lead inputs on the first two 74LS193 counters. The most significant byte of the initial word count value is output when the CPU executes an I/O write instruction to port "BASE+D". The OUTP LEN MSBY/ signal is applied to the load inputs on the other two 74LS193 counters.

If the length register contents do not equal zero, the contents are decremented before each data word is transferred. The LENGTH CNT signal, generated by the DMA transfer control logic, is applied to the down-count input on the first counter. The borrow output from each counter is, in turn, applied to the down-count input of the next counter; essentially creating a 16-bit binary counter out of the four 4-bit counters. The borrow output from the fourth counter constitutes the (LEN REG=Ø)/ signal, which when low (active) indicates that all of the specified data words have been transferred.

The outputs of the four counters are applied to two 74S258 eight-to-four multiplexers. When the CPU executes an I/O read instruction to port "BASE+4", a low level on the RD LEN LSBY/ signal (active) from the address decoding logic enables the least significant byte of the 16-bit word count value (i.e., the contents of the first two counters) through the two multiplexers and onto the system data bus (DATØ/–DAT7/). Similarly, when the CPU executes an I/O read instruction to port "BASE+5", the low (active) level on the RD LEN MSBY/ signal and the high (inactive) level on the RD LEN LSBY/ signal, enable the contents of the other two counters (the most significant byte) through the two multiplexers and onto the system data bus (DATØ/–DAT7/).

8.2.5 MEMORY ADDRESS REGISTER

The memory address register, like the length register, consists of four 74LS193 4-bit counters, as shown on sheet 3 of the module schematic, Figure 8-9. Prior to the beginning of a DMA transfer operation, the memory address register is loaded with the 16-bit address of the first location to be accessed.

The least significant byte of the address is output onto the system data bus (not the system address bus) when the CPU executes an I/O write instruction to port "BASE+E_{16}". The address decoding logic generates the OUTP MA LSBY/ signal which is applied to the load inputs on the first two 74LS193 counters. The most significant byte of the memory address is output when the CPU executes an I/O write instruction to port "BASE+F_{16}". The OUTP MA MSBY/ signal is applied to the load inputs on the other two 74LS193 counters.

If the contents of the length register do not equal zero, the contents of the memory address register are incremented before each data word is transferred. The MEMORY CNT signal, generated by the DMA transfer control logic, is applied to the up-count input on the bottom counter. The carry output from each counter is, in turn, applied to the up-count input of the counter above it. The carry output from the top counter is not used.

The outputs of the four counters feed sixteen 8098 inverters which, when enabled by the SELECTED/

signal from the bus interface control logic (indicating that the DMA Module has control of the system bus), drive the 16 address bits to memory on the system address bus. ADRØ/–ADRF/ (pins P1-43 to P1-58).

These address bits are accompanied by a memory read (MRDC/) or write (MWTC/) command, generated in the DMA transfer control logic. MRDC/ or MWTC/ inform the memory when the address on the bus is valid, as well as specifying the transfer direction. Timing for MRDC/ and MWTC/ is provided in Section 8.2.10.

8.2.6 TAG REGISTER

The tag register consists of four 3404 high-speed latches, as shown on sheet 2 of the module schematic, Figure 8-9.

The tag register is loaded from bits 0–3 of the system data bus (DATØ/–DAT3/) when the CPU executes an I/O write instruction directed to port "BASE+B_{16}". The LOAD TAG REGISTER/ signal from the address decoding logic enables the write inputs on the 3404 latches. The four outputs from the tag register, TAGØ/ (pin J2-40), TAG1/ (pin J2-38), TAG2/ (pin J2-36) and TAG3/ (pin J2-24), are driven by four 7437 NAND gates to the external devices. The tag register is cleared by the system reset signal (SYS RST/), provided that system data bus lines DATØ/–DAT3/ are all false (high).

The tag register can be used at the designer's discretion. For example, each bit in the register can be used as a command line to the external devices. Or, the tag register can be used to expand the maximum number of ports supported by the DMA Module. In this case, the four tag lines could "steer" the data, output with any of the OUTPORT X/ strobes (see Section 8.2.1), to one of 16 devices.

8.2.7 STATUS LOGIC

The status logic consists of two 74S257 multiplexers that gate four internal and four external status lines onto the system data bus during read status operations, as shown on sheet 3 of the module schematic, Figure 8-9.

When the CPU executes an I/O read instruction directed to port "BASE+6", the RD DMAC STAT/ and INPUT EN2/ signals (generated in the address decoding logic), allow the status lines through the two multiplexers and onto the appropriate system data bus lines, as listed in Table 8-3.

The status information on the data bus remains stable until the I/O read command, IORC/, goes false. Figure 8-3 illustrates read status timing.

Table 8-3

STATUS LINES

	STATUS LINE	(PIN)	SOURCE	DATA BUS LINE	(PIN)
External	STATUS0/	(J1-48)	External Device	DAT0/	(P1-73)
	STATUS1/	(J1-46)		DAT1/	(P1-74)
	STATUS2/	(J1-44)		DAT2/	(P1-71)
	STATUS3/	(J1-42)		DAT3/	(P1-72)
Internal	SET INT/		DMA interrupt logic	DAT4/	(P1-69)
	MEM WRT/		DMA transfer ctl logic	DAT5/	(P1-70)
	INTR STAT/		DMA interrupt logica	DAT6/	(P1-67)
	DMA BUSY/		Control register	DAT7/	(P1-68)

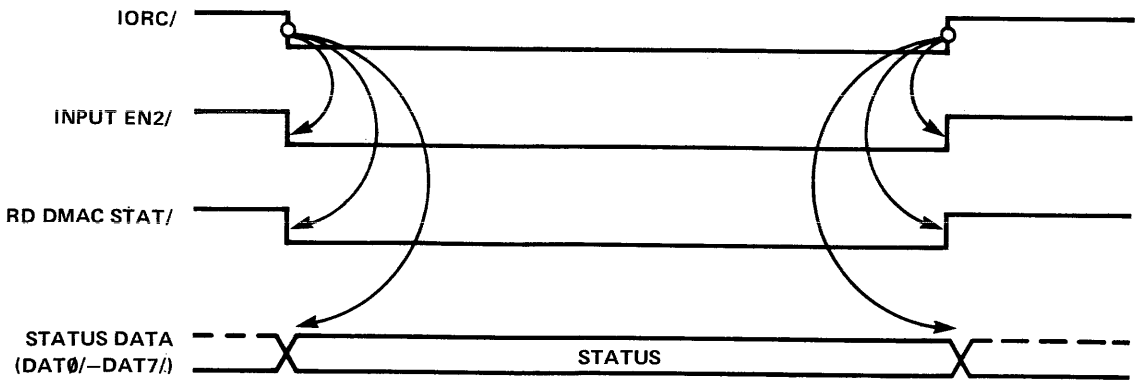


Figure 8-3. Read Status Timing

8.2.8 BUS IN/OUT LOGIC

The bus in/out logic routes data between the data in bus (from external devices), the data out bus (to external devices), the bidirectional system data bus and various registers within the DMA Module. This block consists of an 8212 8-bit I/O port device, sixteen 3404 latches, two 74157 multiplexers and various gating circuits as shown on sheet 3 of the module schematic, Figure 8-9.

Control or address information that is output by the CPU on the system data bus and intended for internal DMA Module registers is inverted and latched into 3404 circuits (by the IOWC/ signal), and applied to two 74157 multiplexers. These multiplexers allow the DMA Module to accept either 8-bit or 16-bit words. When the RD MSBY signal from the DMA transfer control logic is low (false), the low-order data lines (DATØ–DAT7) are multiplexed through to the internal registers. When RD MSBY is high (true), the high-order data lines (DAT8–DATF) are multiplexed through. The DMA transfer control logic (see Section 8.2.10) alternates the level on RD MSBY when a 16-bit operation is indicated by the contents of the control register. You will notice on sheet 3 of the module schematic that data lines DATØ/–DAT5/ are tapped off the sytem bus prior to the 3404 circuits and applied directly to the control and tag registers.

Control or address information that is output by the CPU and intended for control or address registers in an external device are inverted, buffered and multiplexed through the bus in/out logic as described above. The I/O write command (IOWC/) enables eight 7437 NAND gates that drive the byte to the external devices on the data out bus (DATA OUTØ/–DATA OUT7/). One of the OUTPORT strobes, generated in the address decoding logic, informs the appropriate external device that it is to accept the byte in the data out bus.

DMA data that is being transferred from memory to an external device also follows the same path described in the preceding paragraphs. In this case, however, it is the GATE READ DATA/ signal from the DMA transfer control logic which enables the eight 7437 NAND gates that drive the data out bus, and it is the XFER ACKNOWLEDGE/ signal (pin J1-4), also from the DMA transfer control logic (see Section 8.2.10), that strobes the data byte into the external device which issued a transfer request signal (XFER REQ/; pin J1-1). XFER REQ/ is issued for each byte to be transferred to/from an external device. XFER REQ/ also enables the GATE READ DATA/ signal mentioned above.

DMA data that is being transferred from an external device to memory is passed to the DMA Module on the data in bus (DATA INØ/–DATA IN7/). If the module is operating in the 8-bit mode (as defined by the contents of the control register) or if the data byte is the least significant byte of a 16-bit transfer, the data from the data in bus is applied to the eight inputs of an 8212 I/O port device. The MEMORY WRT, ENABLE WRT DATA/ and LD LSBY signals from the DMA transfer control logic (see Section 8.2.10) enable the 8212 section and are driven over the low-order lines of the system data bus (DATØ/–DAT7/). The

generation of MEMORY WRT, ENABLE WRT DATA/ and LD LSBY coincide with the presence of the transfer request signal (XFER RQ/) from the external device. During 16-bit DMA write operations, however, MEMORY WRT and LD LSBY only occur with every other XFER RQ/ signal. The data sent by the external device at this time (via the data in bus) constitutes the most significant byte of the 16-bit data word (see Section 8.2.10).

8.2.9 DMA INTERRUPT LOGIC

The DMA interrupt logic consists of a 7474 D-type flip-flop (interrupt latch), a nine-position rotary switch (interrupt level select) and assorted gating circuits as shown on sheet 2 of the module schematic, Figure 8-9.

The interrupt latch can be set in three different ways:

- The CPU issues a set interrupt command by executing an I/O write instruction to port "BASE+8". (This also sets the SET INT latch.)
- A peripheral device activates its external interrupt line, EX INTERRUPT/ (pin J1-3).
- The DMA Module completes a transfer operation (i.e., when the length register is decremented to zero).

When the CPU executes an I/O write instruction to port "BASE+8", the address decoding logic generates the SET INT/ signal. SET INT/ is saved in the S-R latch, composed of two 7400 OR gates. The output of the S-R latch, SET INT STATUS/ constitutes one of the four internal status lines that can be read by the CPU (see Section 8.2.7). SET INT/ is also inverted and NANDed with DMA BUSY/. If DMA BUSY/ is high (i.e., the DMA Module is not busy), the NAND gate output pre-sets the interrupt latch (A65-10).

If an external device activates its EX INTERRUPT/ line, the interrupt latch is pre-set.

If the DMA Module is currently involved in an operation (i.e., DMA BUSY is true) when the (LEN REG=Ø)/ signal goes true, the interrupt is clocked to the set state.

Setting the interrupt latch, however, does not automatically cause an interrupt request to be asserted. The Q output of the interrupt latch is ANDed with DELAY INT/. Thus, if an external device has activated its delay interrupt line, the interrupt will not be indicated until the delay is removed. The output of this AND gate (A52-6) is inverted and fed to the status logic under the mnemonic INTERRUPT STATUS/. INTERRUPT STATUS/ is one of the four internal status lines (see Section 8.2.7). The output of the AND gate (A52-6) is also applied to one input of a 7438 NAND gate. The other input is the ENABLE INTERRUPT line from the DMA Module's control register (see Section 8.2.3). If interrupts are enabled, the open collector output from this 7438 section is applied to the nine-position interrupt level select switch (S1). The first eight switch positions are tied to the eight interrupt level request lines (INTØ/–INT7/); position 9 is off. The interrupt request to the CPU will be asserted on the level selected by the switch position.

After the CPU services the interrupt request, it will issue a reset interrupt request command (RESET INTERRUPT/) by executing an I/O write instruction to port "BASE+9". RESET INTERRUPT/ resets the interrupt latch, the SET INT latch and the BYTE CNTR latch in the DMA transfer control logic (see Section 8.2.10). RESET INTERRUPT/ also clears the DMA BUSY/ signal, shown on sheet 2 of the module schematic. In addition, the RESET INTERRUPT/ signal is made available to the external devices (via pin J1-14).

8.2.10 DMA TRANSFER CONTROL LOGIC

The DMA transfer control logic is responsible for the "handshaking" exchanges with the peripheral device and memory during all DMA operations. The DMA transfer control logic consists of two 7474 D-type flip-flops, two 74S74 D-type flip-flops, one 74109 J-$\overline{K}$ flip-flop and assorted gating circuits, as shown on sheet 1 of the module schematic, Figure 8-9.

Once a DMA transfer operation has been initiated, the external device will activate its transfer request line (XFER RQ/) whenever it is ready to send a data byte to memory or to receive a byte from memory. The direction of data flow is indicated by

the level on the XFER DIR IN line (pin J1-6) from the external device. A high level on XFER DIR IN specifies that the device expects to receive a data byte from memory; a low level specifies that the device has data to be written to memory. If the XFER DIR IN is left open (high), bit 0 (WRITE) in the control register specifies the direction of data flow.

XFER RQ/ is received at pin J1-1, inverted, buffered and applied to the clock input of the BUS XFER RQ/ latch. This 7474 latch will be clocked to the set state *unless* one of the following is true:

(1) the contents of the control register specify that the current DMA operation is not to involve a data transfer (i.e., if INHIBIT XFER/ from the control register is true);

(2) the contents of the length register are equal to zero (i.e., if (LEN REG=Ø)/ is true);

(3) the first (least significant) byte of a 16-bit word is being input from an external device; or

(4) the second (most significant) byte of a 16-bit word is being output to an external device.

The $\overline{Q}$ output of the 7474 section, BUS XFER RQ/, is fed to the bus interface logic (see Section 8.2.11).

When one of the first two above-mentioned conditions is true (INHIBIT XFER/ or (LEN REG=Ø)/), BUS XFER RQ/ is not generated because no transfer is to occur. However, if any of the four conditions are true when XFER RQ/ occurs, the XFER ACKNOWLEDGE (another 7474 section) latch will be clocked to the set state. When this latch is set, XFER ACKNOWLEDGE/ is asserted at pin J1-4. During all other DMA transfer cycles, the XFER ACKNOWLEDGE/ latch is clocked by RQ ACK/. RQ ACK/ is generated when the DMA Module receives the transfer acknowledge signal (XACK/), returned by memory in response to a memory read (MRDC/) or write (MWTC/) command. Thus, the external device's transfer request is always acknowledged, whether a transfer is actually performed or not. The XFER ACKNOWLEDGE/ latch is pre-reset when XFER RQ/ goes false.

During 16-bit DMA transfer to memory, BUS XFER RQ/ is generated when the external device activates XFER RQ/ in order to send the *second* (most significant) byte to the DMA Module. The entire 16-bit word (two bytes) is then transferred in parallel to memory. The first (least significant) byte was latched into an 8-bit 8212 device (A17) on the previous XFER RQ/.

During 16-bit DMA transfers from memory, BUS XFER RQ/ is generated when the external device activates XFER RQ/ in order to receive the *first* (least significant) byte of data. Memory, in this case, will actually send both data bytes at once. However, the two data bytes will be transferred to the external device separately.

This byte selection process is controlled by the BYTE CNTR latch (a 72S74 latch shown on sheet 1 of the module schematic). During 8-bit transfer operations, the 8-BIT XFER/ signal from the control register keeps the BYTE CNTR latch pre-set (A9-4). During 16-bit transfers, however, the 8-BIT XFER/ signal is false and the latch toggles on the trailing (positive-going) edge of each XFER RQ/ pulse. When the BYTE CNTR latch is clocked to the set state it indicates that the most significant (second) data byte is to be transferred to/from the external device. RD MSBY is generated when the BYTE CNTR latch is set and XFER REQ/ goes true. When the BYTE CNTR latch is clocked to the reset state or when 8-BIT XFER/ is true, it indicates that the first, least significant (or only) data byte is to be transferred to/from the external device. LD LSBY is generated when XFER RQ/ goes true, unless the length register equals zero. RD MSBY and LD LSBY are used to route the data through the bus in/out logic (see Section 8.2.8). The BYTE CNTR latch is pre-reset by a RESET INTERRUPT command (see Section 8.2.9). This ensures that the BYTE CNTR latch is in the proper state for a 16-bit transfer which may follow.

The DMA transfer control logic also generates the read (MRDC/) and write (MWTC/) commands for memory, as well as various read and write signals that are used internally. The Q output of the BUS XFER RQ/ latch (A66-9) feeds two NAND gates. The other input to the first 7400 gate is the result of ANDing XFER DIR IN and WRITE/. This resultant ANDed output is referred to as MEM WRT/. Both XFER DIR IN and WRITE/ are high

during transfers from memory (memory read); thus, MEM WRT/ is false (high) and enables the first 7400 gate (A8-4). MRD RQ/ is the output from this gate (it is true during memory read operations). The AND of XFER DIR IN and WRITE/ is also inverted (referred to as MEMORY WRT; active-high) and applied to the other input of the second 7400 NAND gate. MWT RQ/ is the output from this gate (it is true during memory write operations). If the DMA Module's bus interface logic has gained control of the system bus (see Section 8.2.11), MRD RQ/ and MWT RQ/ are gated through to pins P1-19 and P1-20 and driven to memory as MRDC/ and MWTC/, respectively. MRD RQ/ is also available to the bus in/out logic, where it causes the data being read from memory to be latched into 3404 circuits (see Section 8.2.8).

When MEMORY WRT is true (high) it is also NANDed with the DMA SELECTED signal from the bus interface logic to produce ENABLE WRT DATA/. When MEM WRT/ is false (high), it is NANDed with XFER RQ to produce GATE READ DATA/. MEM WRT/ is also inverted and made available to the external device as XFER DIR OUT (pin J1-2). XFER DIR OUT is high during memory read operations and low during memory write operations. MEMORY WRT, MEM WRT/, ENABLE WRT DATA/ and GATE READ DATA/ are all made available to the bus in/out logic, where they are used to gate data on or off the system data bus (see Section 8.2.8).

Figures 8-4 and 8-5 illustrate the relative timing between DMA transfer control logic signals during memory read cycles (memory-to-external device) for 8 and 16-bit transfers, respectively. Figures 8-6 and 8-7 illustrate timing during memory write cycles (external device-to-memory) for 8 and 16-bit transfers, respectively.

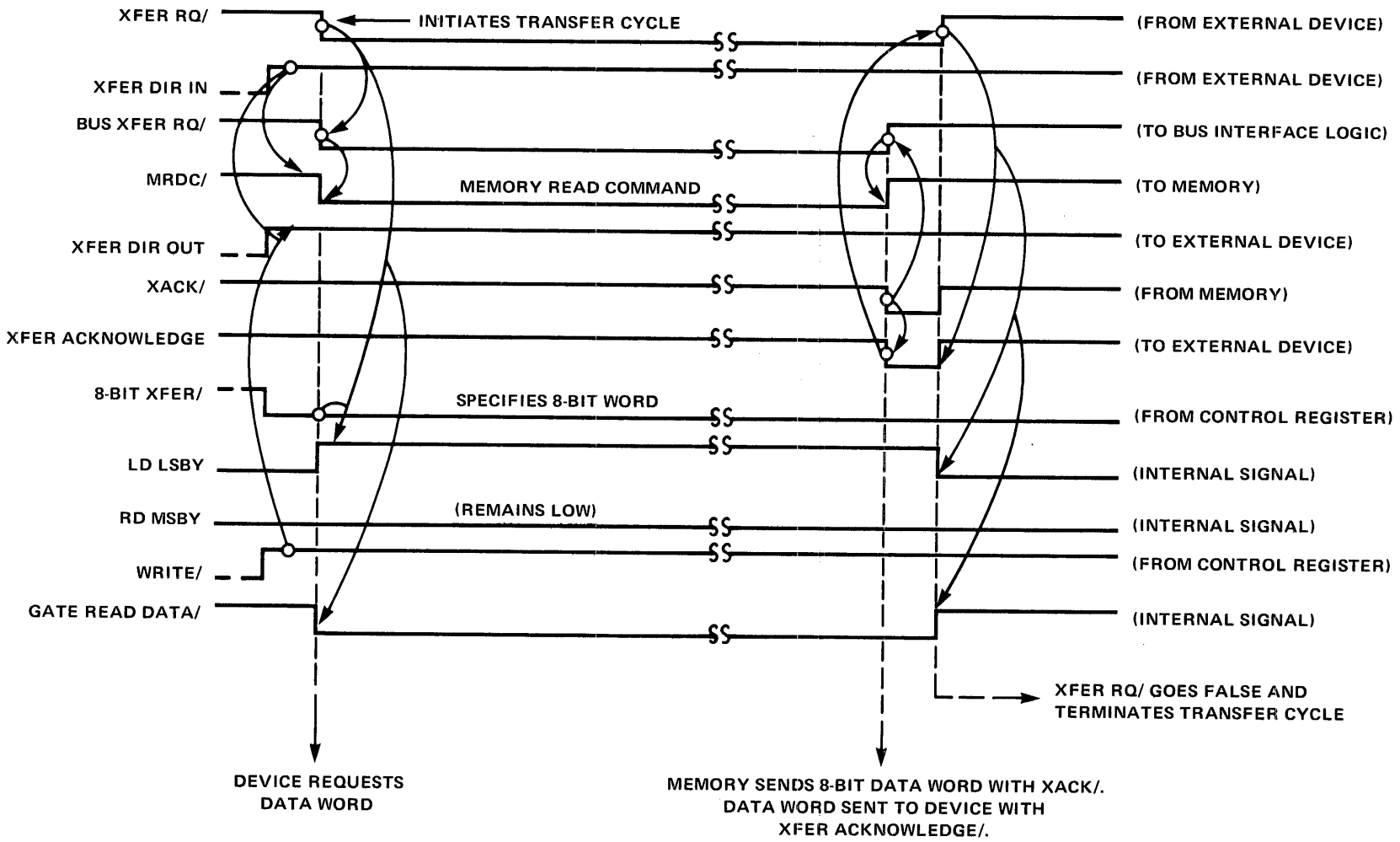
8.2.11 BUS INTERFACE LOGIC

The bus interface logic is responsible for requesting and maintaining control of the sytem bus when the DMA Module is conducting a data transfer. The bus interface logic consists of two 74S74 D-type flip-flops, two 74109 J-$\overline{K}$ flip-flops and assorted gating circuit, as shown on sheet 1 of the module schematic, Figure 8-9.

When the DMA transfer control logic determines that a data transfer is to be performed by the DMA Module, it issues the BUS XFER RQ/ signal to the bus interface logic. BUS XFER RQ/ is inverted and applied to the D input of the bus request latch (74S74 flip-flop). The bus request latch is subsequently clocked to the set state, on the positive-going edge of the next bus clock pulse (BCLK/). The $\overline{Q}$ output (BREQ/) is asserted at pin P1-18. BREQ/ specifies that the DMA Module requires use of the system bus. Logic on some other module will resolve all bus requests according to priority. In the INTELLEC MDS System, the Front Panel Control Module provides an eight-level, parallel bus priority scheme.

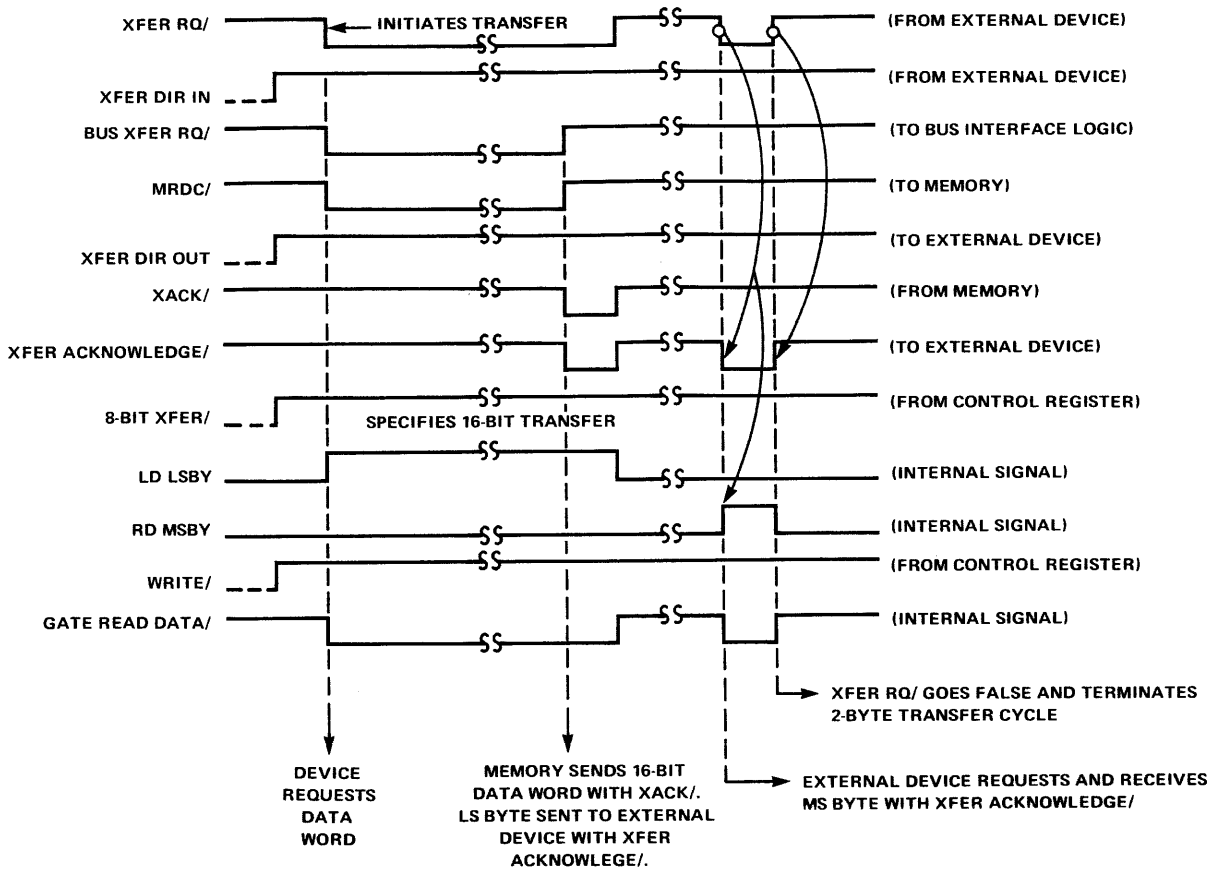
When the bus is available for use by the DMA Module, the BPRN/ signal (pin P1-15) will be true. BPRN/ is inverted and applied to one of three inputs on a 74H11 AND gate (A25-2). The other two inputs are supplied by the Q output from the bus request latch and the bus busy signal (BUSY/). Thus, the 74H11 gate will be activated if PBRN is true, the DMA Module is currently requesting use of the bus (BREQ is true) and the bus is not already in use (BUSY/ is false). The output of this 74H11 section (A25-12) feeds the J input of the bus busy latch (a 74109 section). The bus busy latch is clocked set by the positive-going edge of the next bus clock pulse (BCLK/). The Q output (DMA SELECTED) informs the DMA transfer control logic that the module now has control of the system bus. The $\overline{Q}$ output (EN MEM ADR/) enables the 74125 driver which asserts the bus busy signal (BUSY/) at pin P1-17. EN MEM ADR/ also enables the sixteen 8098 inverters which drive the contents of the memory address register on the system address bus (ADRØ/–ADRF/). This address is available to memory. DMA SELECTED is also fed to the J and $\overline{K}$ inputs on another 74109 flip-flop. The next BCLK/ pulse clocks this section to the set state. The $\overline{Q}$ output, in turn, enables the two 74125 drivers which enable the memory read (MRDC/) or write (MWTC/) command onto the system bus (pins P1-19 and P1-20, respectively). Thus, the appropriate memory command (see Section 8.2.10) is gated onto the system bus one bus clock period after the memory address is enabled.

Normally, the DMA Module will only retain control of the bus for the transfer of one data word. An



NOTE: THIS DIAGRAM ASSUMES THAT THE DMA MODULE HAS CONTROL OF THE SYSTEM BUS (ALSO REFER TO FIGURE 8-8).

Figure 8-4. DMA Transfer from Memory (8-Bit Data Word)



NOTE: THIS DIAGRAM ASSUMES THAT THE DMA MODULE HAS CONTROL OF THE SYSTEM BUS (ALSO REFER TO FIGURE 8-8).

Figure 8-5. DMA Transfer from Memory (16-Bit Data Word)

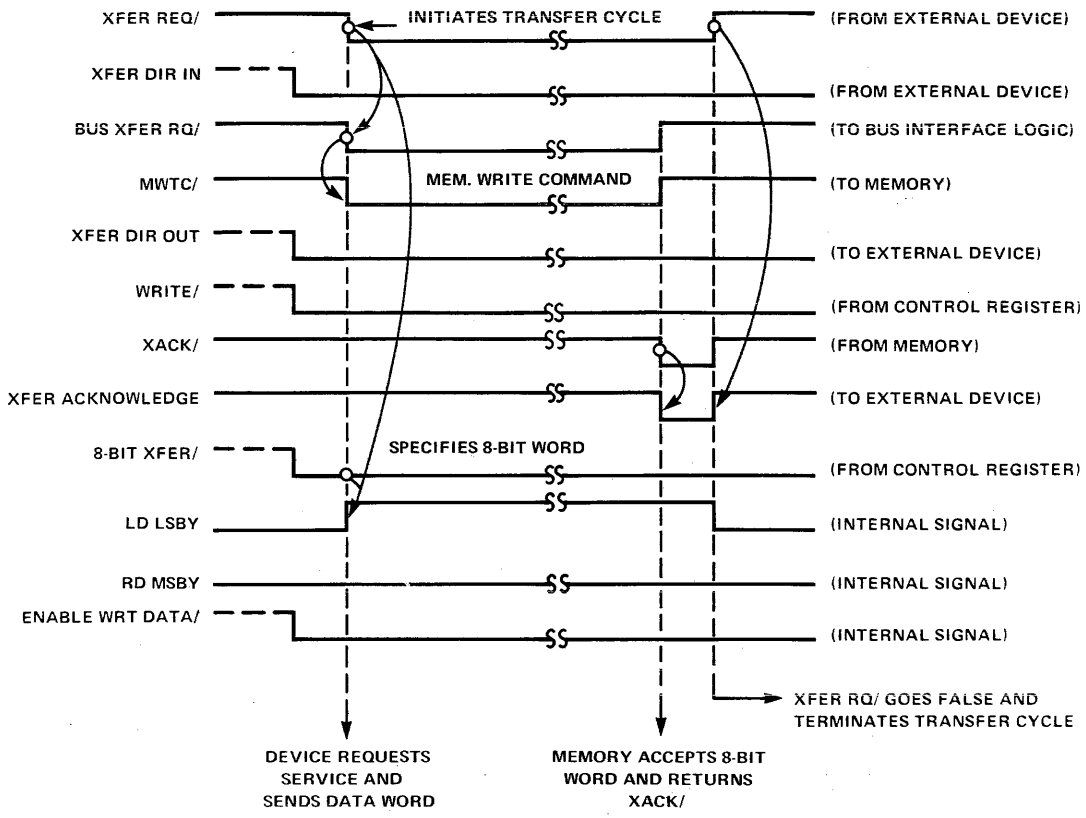
active (low) level on the BUSY/ line disables the 74H11 gate, feeding the J input of the bus busy latch; consequently, the bus busy latch is clocked reset by the next BCLK/ pulse, unless the $\overline{K}$ input is high. If the $\overline{K}$ input is high, the flip-flop does not change states; that is, it remains set and the DMA Module retains control of the bus. The DMA Module will retain control of the bus (i.e., the $\overline{K}$ input will be high) if any of the following conditions are true when the positive-going edge of BCLK/ occurs:

- OVERIDE (from the control register) is true,
- BPRN/ and BUS XFER RQ/ are both true, or
- the RETAIN BUS/ latch is in the reset state.

OVERIDE/ will be true if the appropriate bit (bit 5) in the control register was set by the CPU and an enabled DMA interrupt request is not currently active (INTERRUPT STATUS/ is false). OVERIDE/ allows the DMA Module to conduct "burst" mode transfers.

If the DMA transfer control logic issues a new BUS XFER RQ/ signal while BPRN/ is true, but before the next BCLK/ has occurred, the bus interface logic will retain control of the bus for at least one more transfer. This prevents the DMA Module from initiating a transfer, then losing the bus before the transfer is actually performed.

The RETAIN BUS/ latch provides a similar safeguard. It prevents the DMA Module from relinquishing control of the bus after it has issued a read or write command to memory, but before



NOTE: THIS DIAGRAM ASSUMES THAT THE DMA MODULE HAS CONTROL OF THE SYSTEM BUS (ALSO REFER TO FIGURE 8-8).

Figure 8-6. DMA Transfer to Memory (8-Bit Data Word)

memory has actually completed the transfer. If the DMA Module has control of the bus (SELECTED/ is true) the RETAIN BUS/ latch is clocked when the memory read (MRD RQ/) or memory write (MWT RQ/) request signal (from the DMA transfer control logic) or the bus request signal (BREQ/) is generated. The latch will be clocked to the reset state if OVERIDE/ or BPRN/ is true. Otherwise, the latch will remain reset. The Q output (low when the latch is reset) enables the MRD RQ/ or MWT RQ/ signal through to the MRDC/ or MWTC/ driving circuit. The Q output also asserts a high level on the K input of the bus busy latch (as mentioned above), forcing the bus interface logic to retain control of the bus. When MRD RQ/, MWT RQ/ and BREQ/ are all false (high), the RETAIN BUS/ latch is pre-set.

The DMA Module's bus interface logic is primarily controlled by the state of the bus priority in (BPRN/) signal. When BPRN/ is true, the module can gain or retain control of the bus, and when BPRN/ is false, the module will relinquish control of the bus, unless override has been invoked. BPRN/ may be generated by a central *parallel* priority network; in the INTELLEC system such a network is included on the Front Panel Control Module. BPRN/ may also be generated and transmitted in *serial*. In such a case, BPRN/ is captured by the highest priority module requiring control of the bus. Those modules that do not require the bus accept BPRN/ and pass BPRO/ (bus priority out) on to the next module on the bus (via pin P1-16). Thus, a module's priority in a serial network is dependent on its relative position on the bus.

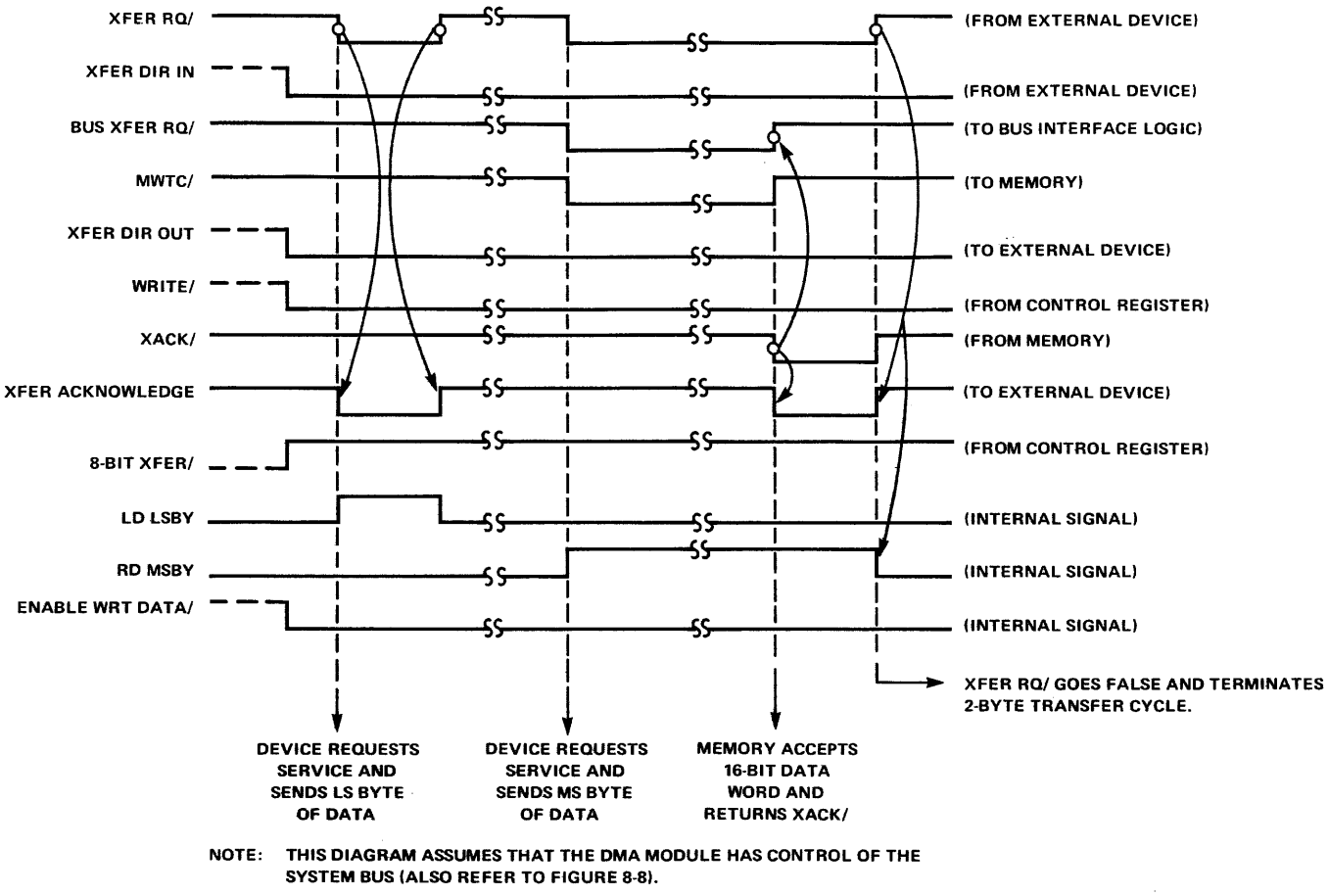


Figure 8-7. DMA Transfer to Memory (16-Bit Data Word)

BPRO/ is generated if BPRN/ is true and BREQ is false (i.e., the DMA Module is not requesting use of the system bus).

Figure 8-8 illustrates timing within the bus interface logic.

8.2.12 DMA MODULE SCHEMATIC

Figure 8-9 provides a complete schematic drawing (3 sheets) of all logic on the DMA Module.

8.3 UTILIZATION: DMA MODULE

This section provides information on utilization of the DMA Module.

8.3.1 INSTALLATION

In installing the DMA Module, the user must take account of:

(a) environmental extremes
(b) mounting considerations
(c) electrical connections
(d) power requirements
(e) signal requirements
(f) base address selection
(g) interrupt level selection

Environment

Temperature extremes can cause instability, or result in permanent damage to the circuits on the

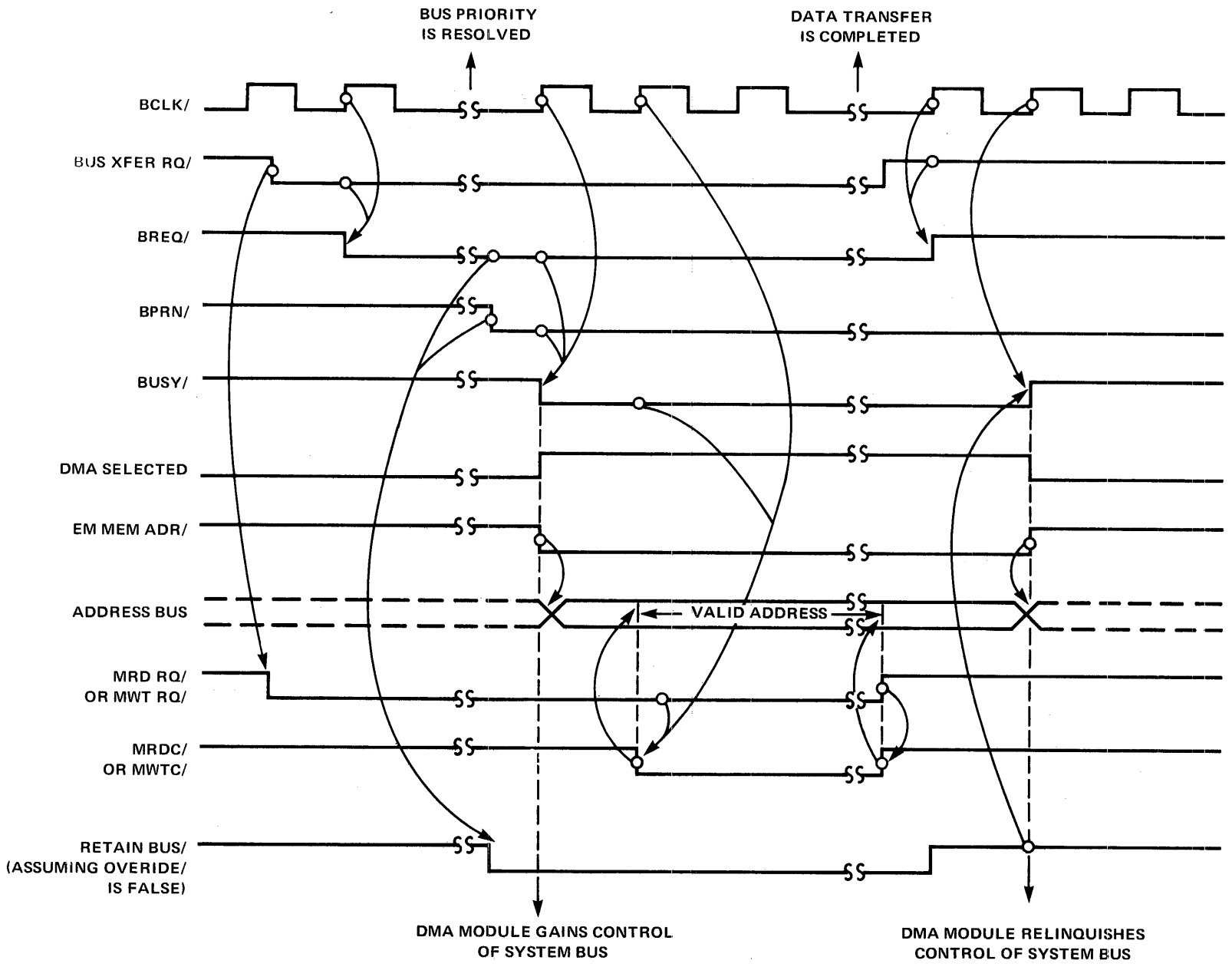


Figure 8-8. Bus Interface Timing

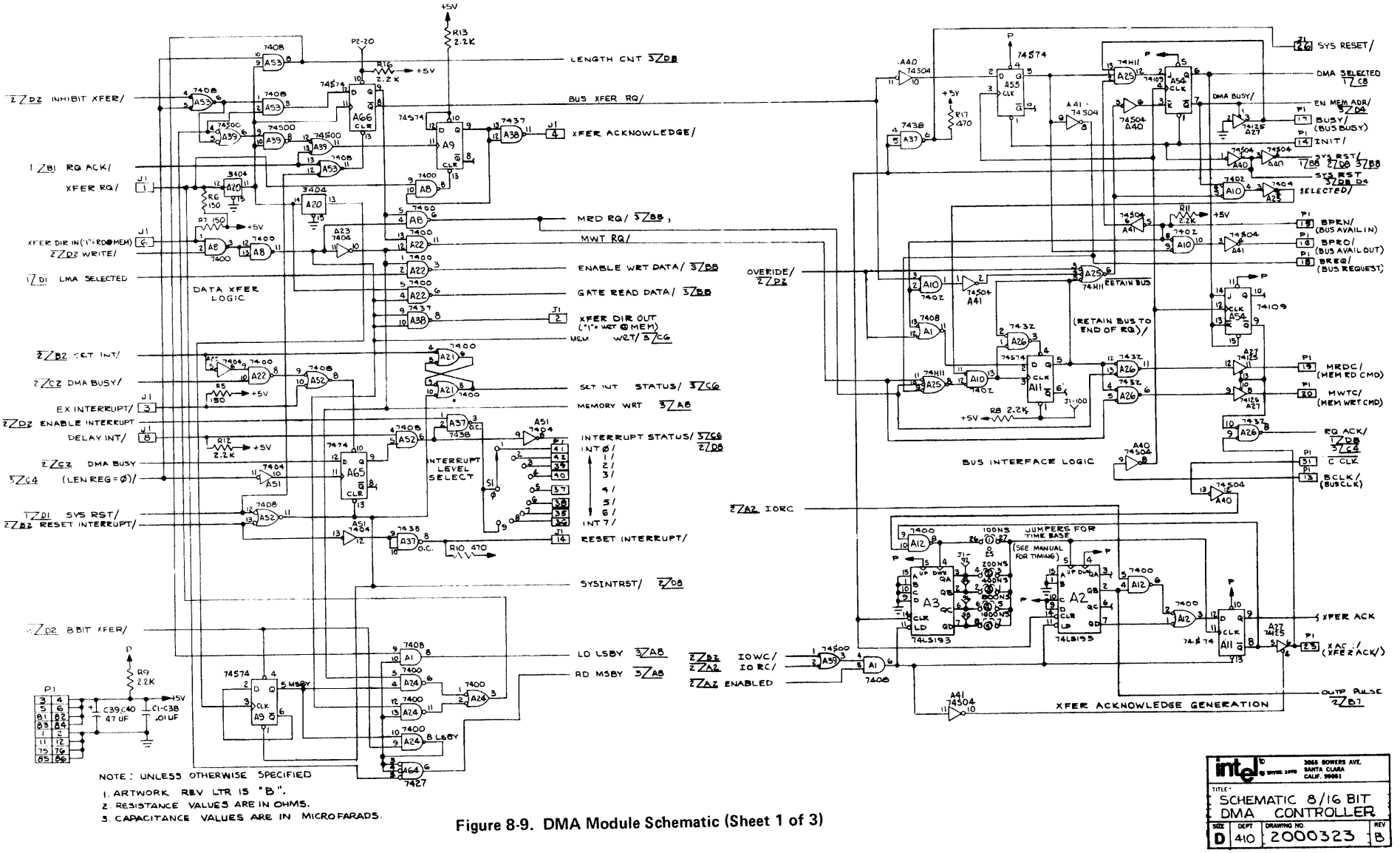


Figure 8-9. DMA Module Schematic (Sheet 1 of 3)

intel	3065 BOWERS AVE. SANTA CLARA CALIF. 95051		
TITLE: SCHEMATIC 8/16 BIT DMA CONTROLLER			
SIZE: D	DEPT: 410	DRAWING NO.: 2000323	REV: B

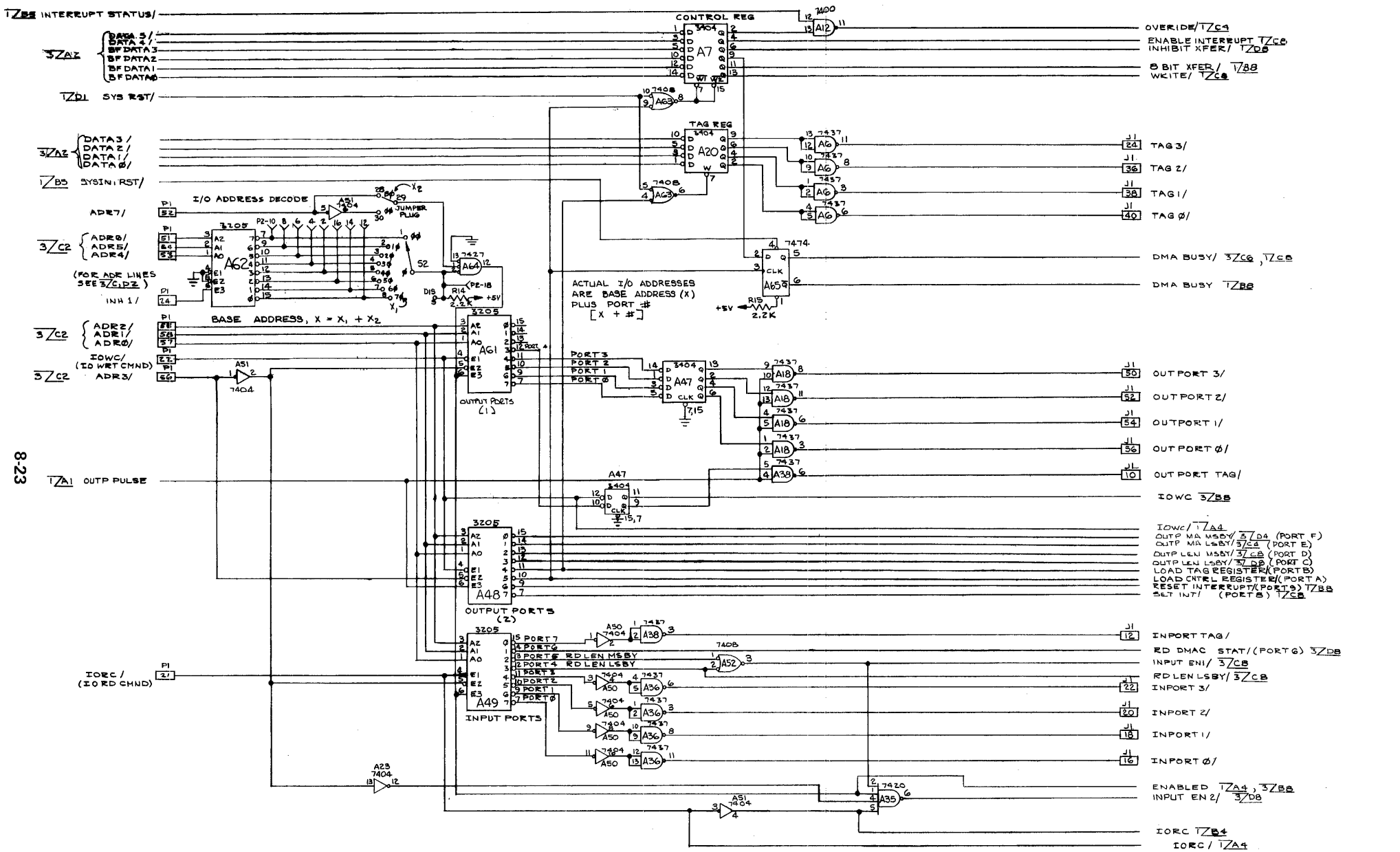


Figure 8-9. DMA Module Schematic (Sheet 2 of 3)

SCALE: ~	SIZE D	DEPT 410	DRAWING NO. 2000323	REV B
SHEET 2 OF 3				

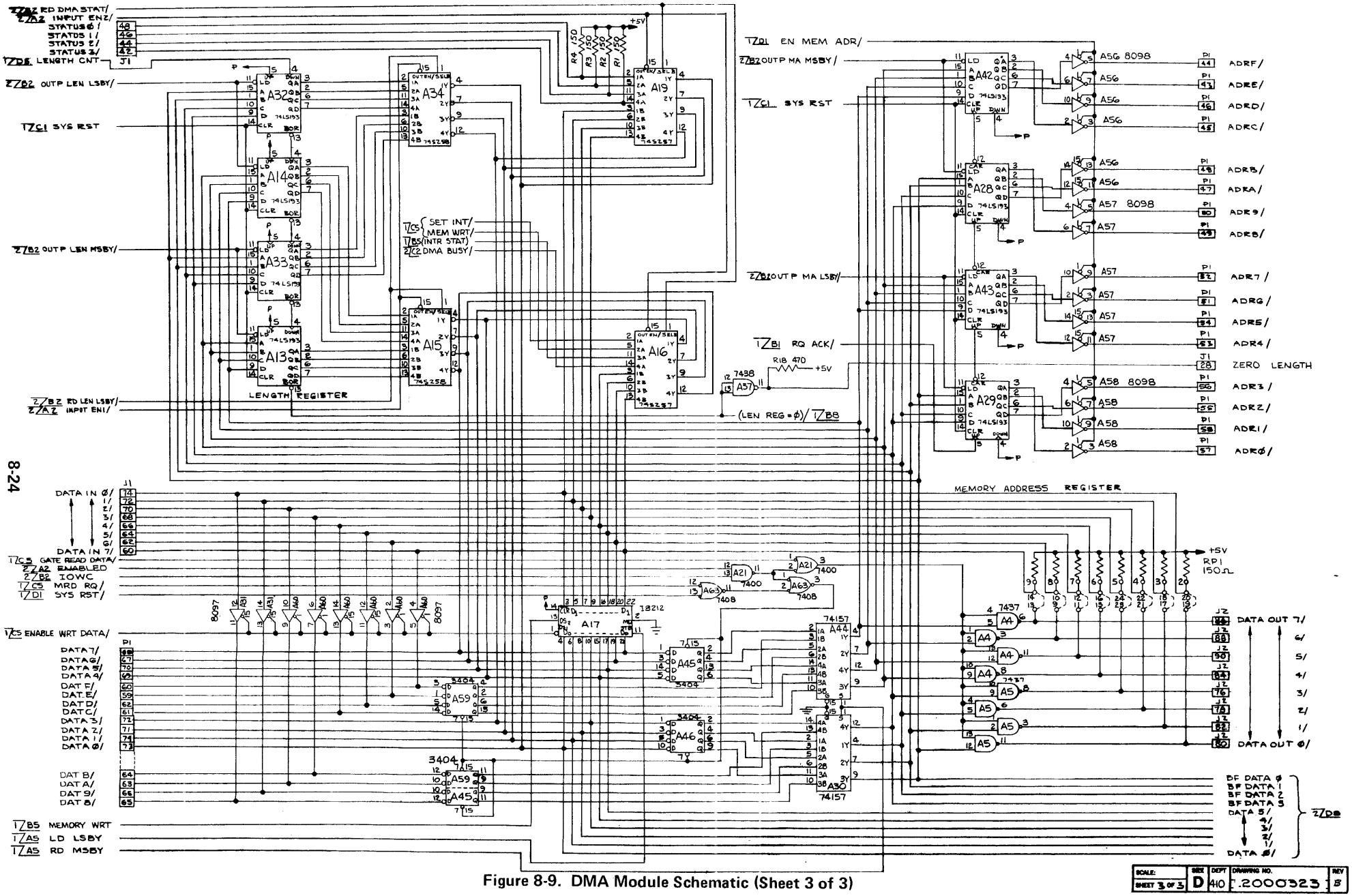


Figure 8-9. DMA Module Schematic (Sheet 3 of 3)

module. Ambient temperatures must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conductive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity should not exceed 90%, noncondensing.

Mounting

Avoid locating the module near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections on the board, resulting in abnormally high noise levels or outright failure of the assembly.

Dimensions of the module are 12-in. X 6.75-in. Be sure to allow enough additional clearance to ensure adequate cooling.

The module is designed to plug directly into three standard, double-sided PC edge connectors. An 86-pin connector and a 60-pin auxiliary connector are located on one edge of the board; a 100-pin connector is on the opposite edge. The connectors can serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the assembly be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the board, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the components on the module.

Electrical Connections

The DMA Module communicates with the motherboard and, consequently, the rest of the system, through a standard 86-pin, double-sided PC edge connector (P1), 0.156-in. contact centers, as shown in Figure 8-10. Control Data VPB01E43A00A1 is one suitable type of connector. Pin allocations on this connector are given in Table 8-4 of Section 8.3.2. The module can also communicate with other modules in the system, through the auxiliary 60-pin, double-sided PC edge connector (P2), 0.1-in. contact centers (see Figure 8-10). Pin allocations for this connector (primarily test points) are listed in Table 8-5. The module transfers information to/from the peripheral devices via a 100-pin, double-sided PC edge connector (J1) which attaches to the edge opposite that of the other two connectors. This connector has 0.1-in. contact centers. Viking 3VH50/1JN5 is one suitable type of connector for communicating with the peripheral devices. Pin allocations for this connector are given in Table 8-6.

The DMA Module requires +5 VDC power.

Refer to the pin list in Table 8-4 of Section 8.3.2 for power connections.

Signal Requirements

All data and control functions appearing at the module edge connectors are at TTL levels. Electrical characteristics of the signal inputs and outputs, as well as power inputs, are given in Section 8.4.

Signal descriptions and connector pin allocations are given in Section 8.3.2.

Base Address Selection

The user must assign a base address to the DMA Module. The base address is defined by the four most significant bits of the 8-bit I/O port address. The four least significant bits, then, define 16 unique addresses within the range defined by each base address. When the CPU accesses the DMA module by executing an I/O instruction, the base address (BASE) selects the proper DMA Module, while the four low-order address bits select one of the internal DMA registers or I/O ports, as described in Section 8.2.1. For example, an output instruction to port "BASE+B_{16}" loads the tag register on the DMA Module, while an input instruction to address "BASE+3" reads the data byte at input port 3.

A base address is assigned by connecting the proper poles of the X2 jumper plug (28-29-30) and by positioning the X1 rotary switch (S2), shown on sheet 2 of the module schematic. Table 8-1 in

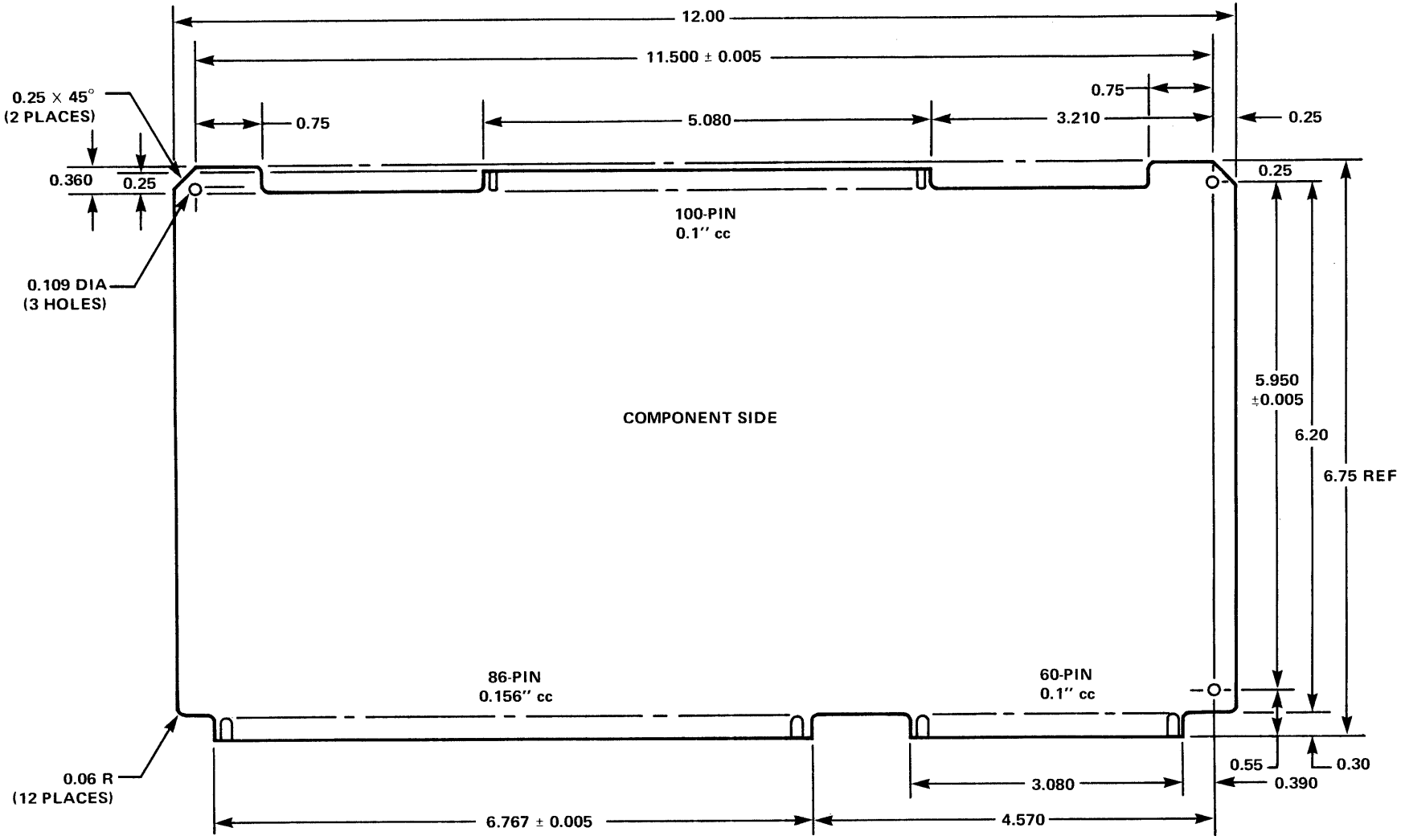


Figure 8-10. DMA Module Connectors

Section 8.2.1 lists the various base addresses that can be selected by connecting the X2 jumper plug and positioning the X1 switch.

Interrupt Level Selection

The user can assign the DMA Module's interrupt request line to any one of eight interrupt priority levels. (INTØ/–INT/7) by moving the interrupt level select switch (S1) to the desired position. This nine-position rotary switch is shown on sheet 1 of the module schematic. The nine switch positions are associated with the following priority levels:

SWITCH POSITION	INTERRUPT PRIORITY LINE	RELATIVE PRIORITY (INTELLEC MDS SYSTEM)
1	INTØ/	Highest
2	INT1/	
3	INT2/	
4	INT3/	
5	INT4/	
6	INT5/	
7	INT6/	
8	INT7/	Lowest
9	OFF	(No Interrupt Request)

8.3.2 PIN LISTS: DMA MODULE

The following secition provides connector pin allocations on the DMA Module. The pins and their designated signal functions for the 86-pin connector (P1) are listed in Table 8-4. The same information for the 60-pin auxiliary connector (P2) is listed in Table 8-5. Pin and signal information for the 100-pin peripheral connector (J1) is given in Table 8-6.

8.4 OPERATING CHARACTERISTICS: DMA MODULE

The AC and DC characteristics of all major signals that appear at the edge connectors is listed in this section.

8.4.1 AC CHARACTERISTICS

AC characteristics are given in Tables 8-7a, 8-7b and 8-8.

8.4.2 DC CHARACTERISTICS

DC characteristics are given in Table 8-8. Power requirements are cited below:

	TYP	MAX
V_{CC} +5 VDC ±5%	2.7A	3.35A

Table 8-4

P1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	GND	Ground
2	GND	
3	+5 VDC	Power inputs
4	+5 VDC	
5	+5 VDC	
6	+5 VDC	
7		
8		
9		
10		
11	GND	Ground
12	GND	
13	BCLK/	Bus clock (9.8304 MHz)
14	INIT/	System reset
15	BPRN/	Bus priority in
16	BPRO/	Bus priority out
17	BUSY/	Bus busy
18	BREQ/	Bus request
19	MRDC/	Memory read command
20	MWTC/	Memory write command
21	IORC/	I/O read command
22	IOWC/	I/O write command
23	XACK/	Acknowledge transfer
24		
25		
26		
27		
28	Zero Length	Indicates Len Reg = ØØØØ
29		
30		
31	CCLK/	Common clock (9.8304 MHz)
32		
33		
34		
35	INT6/	Interrupt requests
36	INT7/	
37	INT4/	
38	INT5/	
39	INT2/	
40	INT3/	
41	INTØ/	
42	INT1/	
43	ADRE/	Address bus

PIN	SIGNAL	FUNCTION
44	ADRF/	Address bus
45	ADRC/	
46	ADRD/	
47	ADRA/	
48	ADRB/	
49	ADR8/	
50	ADR9/	
51	ADR6/	
52	ADR7/	
53	ADR4/	
54	ADR5/	
55	ADR2/	
56	ADR3/	
57	ADRØ/	
58	ADR1/	
59	DATE/	Data bus
60	DATF/	
61	DATC/	
62	DATD/	
63	DATA/	
64	DATB/	
65	DAT8/	
66	DAT9/	
67	DAT6/	
68	DAT7/	
69	DAT4/	
70	DAT5/	
71	DAT2/	
72	DAT3/	
73	DATØ/	
74	DAT1/	
75	GND	Ground
76	GND	
77		
78		
79		
80		
81	+5 VDC	Power inputs
82	+5 VDC	
83	+5 VDC	
84	+5 VDC	
85	GND	Ground
86	GND	

Table 8-5

P2 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1		TEST POINTS
2	40 ADR/	
3		
4	30 ADR/	
5		
6	20 ADR/	
7		
8	10 ADR/	
9		
10	00 ADR/	
11		
12	70 ADR/	
13		
14	60 ADR/	
15		
16	50 ADR/	
17		
18	DIS ADR	
19		
20	SET BUS XFER RQ	
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		

PIN	SIGNAL	FUNCTION
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		

Table 8-6

J1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	XFER RQ/	Transfer request
2	XFER DIR OUT	Transfer direction out
3	EX INTERRUPT/	External interrupt
4	XFER ACKNOWLEDGE/	Transfer acknowledge
5		
6	XFER DIR IN	Transfer direction in
7		
8	DELAY INT/	Delay interrupt
9		
10	OUTPORT TAG/	Output strobe
11		
12	INPORT TAG/	Input strobe
13		
14	RESET INTERRUPT/	Reset interrupt
15		
16	INPORTØ/	Input strobes
17		
18	INPORT1/	
19		
20	INPORT2/	
21		
22	INPORT3/	
23		
24	TAG3/	Tag register, bit 3
25		
26	SYS RESET/	System reset
27		
28		
29		
30		
31		
32		
33		
34		
35		
36	TAG2/	Tag register, bit 2
27		
38	TAG1/	Tag register, bit 1
39		
40	TAGØ/	Tag register, bit 0
41		
42	STATUS 3/	External status, bit 3
43		
44	STATUS 2/	External status, bit 2
45		
46	STATUS 1/	External status, bit 1
47		
48	STATUS 0/	External status, bit 0
49		
50	OUTPORT3/	Output strobes

Table 8-6

J1 CONNECTOR PIN LIST (continued)

PIN	SIGNAL	FUNCTION
51		
52	OUTPORT2/	Output strobes
53		
54	OUTPORT1/	
55		
56	OUTPORTØ/=	
57		
58		
59		
60	DATA IN7/	Data input bus (from device)
61		
62	DATA IN6/	
63		
64	DATA IN5/	
65		
66	DATA IN4/	
67		
68	DATA IN3/	
69		
70	DATA IN2/	
71		
72	DATA IN1/	
73		
74	DATA INØ/	
75		
76	DATA OUT3/	Data output bus (to device)
77		
78	DATA OUT2/	
79		
80	DATA OUTØ/	
81		
82	DATA OUT1/	
83		
84	DATA OUT4/	
85		
86	DATA OUT7/	
87		
88	DATA OUT6/	
89		
90	DATA OUT5/	
91		
92	200 ns	Test point
93		
94	400 ns	
95		
96	800 ns	TEST POINTS
97		
98	1600 ns	
99		
100	ASSERT RETAIN BUS/	

Table 8-7a.

DMA CONTROLLER INTELLEC® BUS MASTER AC CHARACTERISTICS

PARAMETER	OVERALL		CONTINUOUS BUS CONTROL (OVERRIDE)		EXCHANGE OF BUS CONTROL		DESCRIPTION OUTPUT LIMITS	REMARKS
	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)	MIN (ns)	MAX (ns)		
t_{AS}	50		50		$t_{CY}-43$		Address setup time to command	[3] $t_{CY} = t_{BCY}$ (Min Per Apl)
t_{AH}	$t_{XKO}+9$		$t_{XKO}+9$		$t_{XKO}+9$		Address hold time from command	See t_{XKO} below
t_{DS}	[3] $t_{BCY}-59$		[1] 50		[3] $t_{BCY}-59$		Data setup time to command, write	[1] $t_{EDSW}-11_{ns}$ (Table 8-7b)
t_{DHW}	[2] 50		[2] 50		Fdu		Data hold time from command, write	[2] $t_{EDHW}-73.5_{ns}$ (Table 8-7b)
t_{DD}							Data delay during memory write	Data always valid during command (t_{DS})
t_{SEP}	260		260				Command separation	
t_{WC}	$t_{ACC}+15$		$t_{ACC}+15$		$t_{ACC}+15$		Command width	Assuming XACK/ t_{ACC}=delay
t_{DBS}	9	33			9	33	Bus sample to exchange initiation	Sample point = BCLK/↓
t_{BS}	-29	$76+t_{BCY}$			-29	$76+t_{BCY}$	Bus sampling point delay	
t_{DB}		59				59	Data and address turn on delay	
t_{DRQ}		13.5				13.5	Bus request delay	Fig 8-7b
t_{DBY}		57.5				57.5	Bus busy turn on delay	Fig. 8-7b
t_{DBYF}		32.5				32.5	Bus busy turn off delay	Fig. 8-7b
t_{DBPO}		26.5				26.5	BPRO/ serial delay from BPRN/	Fig. 8-7b
t_{XKCO}	15	104.5	15	104.5	15	104.5	Command turn off delay from XACK/	
							Input Requirements	
t_{XKD}	1		1		1		XACK delay from valid read data	
t_{XKO}		70		70		70	XACK turn off delay	
t_{BCY}	100		100		100		Bus clock cycle time	Fig. 8-7b
t_{BW}	25				25		Bus clock low and high periods	Fig. 8-7b
t_{DBPN}		36.5				36.5	Priority input setup time	Fig. 8-7b
t_{DHR}	23		23		23		Data hold from read command	
t_{ACC}		System Timeout		System Timeout		System Timeout	XACK delay from command	
t_{CCY}	100						Comm. clock cycle time	
t_{CW}	25						Com. clock low and high periods	

Table 8-7b

DMA CONTROLLER EXTERNAL AC CHARACTERISTICS

PARAMETER	OVERALL LIMITS$_{ns}$		CONTINUOUS BUS CONTROL (OVERRIDE)		EXCHANGE OF BUS CONTROL		DESCRIPTION OUTPUT LIMITS	REMARKS
	MIN	MAX	MIN	MAX	MIN	MAX		
t_{EXKO}		40.5		40.5		40.5	XFER ACKN/ to XFER RQ/ OFF	
t_{EACC}		159.5 + t_{ACC}		159.5 + t_{ACC}		122 + t_1	Memory Access – XFER RQ to XFER ACKN/	t_1=*t_{BUSMAX} +3t_{BCY}+ t_{ACC}
t_{ECY}	t_{EACC}+ t_{ERQSEP}						XFER Cycle	
t_{EXKO}		40.5		40.5		40.5	XFER ACKN/ OFF from XFER RQ/ RD data	
t_{EXKD}	-39		-39		-39		XFER ACKN/ delay from valid RD data	(=39ns + t_{XKD} (slave))
t_{EDHR}	10		10		10		Valid RD data after XFER RQ/ OFF	
t_{EXINTD}		93					EXT INT to INTx/ delay	
t_{EDI}		46					Delay INT/ to INTx/	
$t_{EXDI/O}$		59					XFERDIRIN to XFERDIROUT delay	
t_{DOTAG}	nt_{CCY}	190.5+ $2nt_{CCY}$					IOWC/ to valid TAGx/, OUTPORTx, XFER DIROUT	N=1,2,4,8,16 jumper selectable
t_{EDSO}	nt_{CCY} -133						Output Data Set-up to OUTPORTx/ STROBE	
t_{ESTB}	13 + $2nt_{CCY}$	122 + $2nt_{CCY}$					OUTPORTx/ STROBE width	
t_{EDHO}	nt_{CCY}						Output Data Hold from OUTPORTx/ STROBE	
t_{DII}	8	55					IORC/ to INPORTx/ delay	
t_{INPORT}	$2nt_{CCY}$ -55						INPORTx/ width	
t_{EDSW}	58		58		0		INDATA and XFERDIRIN set-up to XFER RQ/	
t_{ERQ}	t_{EACC}		t_{EACC}				XFER RQ/ width	t_{ERQ} Min = t_{EACC} t_{EXKCO} Min
t_{ERQSEP}	181.5		181.5				XFER RQ/ Separation	
t_{EDHW}	123.5		123.5				INDATA XFERDIRIN hold time after XFER RQ/	
t_{EXKCO}	0		0		0		XFER ACKN/ to XFER RQ/ OFF	
t_{EACCI}		$2n_{TC}$ -115					DATAINx/ valid from INPORTx/	n=1,2,4,8,16 jumper selectable
		85						t_{CCY}=100ns,n=1
		285						t_{CCY}=100ns,n=2
		685						t_{CCY}=100ns,n=4
		1485						t_{CCY}=100ns,n=8
		3085						t_{CCY}=100ns,n=16

Table 8-7b

DMA CONTROLLER EXTERNAL AC CHARACTERISTICS (continued)

PARAMETER	OVERALL LIMITS$_{ns}$		CONTINUOUS BUS CONTROL (OVERRIDE)		EXCHANGE OF BUS CONTROL		DESCRIPTION OUTPUT LIMITS	REMARKS
	MIN	MAX	MIN	MAX	MIN	MAX		
t_{EDHI} t_{EXTINT}	0 46		0		0		DATAINx/ hold from INPORTy/ EXTINT/ width	*t_{BUSMAX}= longest time a BUS master will keep BUSY/ true.

Table 8-8

DMA CONTROLLER INTELLEC® BUS SLAVE AC CHARACTERISTICS

PARAMETER	OVERALL		DESCRIPTION OUTPUT LIMITS	REMARKS
	MIN	MAX		
t_{DHR}	0		REFER TO FIGURE 8-12	
t_{XKD}	50			
t_{XKO}	40.5			
INPUT REQUIREMENTS				
t_{AS}	33		REFER TO FIGURE 8-12	
t_{AH}	0.5			
t_{DS}	0			
t_{DHW}	50			
t_{WC}	t_{ACC}			
t_{CSEP}	50			
t_{ACCW}		$5nt_{CCY}$ +78.5		n=1,2,4,8,16
t_{ACCR}		$3nt_{CCY}$ +78.5		n=1,2,4,8,16
t_{XKCO}	0			

Table 8-9

INTELLEC® BUS DMA DC CHARACTERISTICS

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST DESCRIPTION	PARAMETER			
				MIN	TYP	MAX	UNITS
ADRØ/ADRF/ ADDRESS	V_{OL}	Output low voltage	I_{OL} = 32 mA			0.4	V
	V_{OH}	Output high voltage	I_{OH} = -5.2 mA	2.4			V
	C_L	Capacitive Load				15	pF
MRDC/,MWTC/ MEMORY COMMAND	V_{OL}	Output low voltage	I_{OL} = 16 mA			0.4	V
	V_{OH}	Output high voltage	I_{OH} = -2 mA	2.4	3.1		V
	I_{LH}	Output leakage high	HIGH Z V_0 = 2.4 V			40	μA
	I_{LL}	Output leakage low	HIGH Z V_0 = 0.4 V			-40	μA
	C_L	Capacitive load				15	pF
IORC/,IOWC/ I/O COMMANDS	V_{IL}	Input low voltage				0.8	V
	V_{IH}	Input high voltage		2			V
	I_{IL}	Input current at V_{IL}	V_{IL} = 0.5 V			-3.85	mA
	I_{IH}	Input current at V_{IH}	V_{IH} = 2.7 V			170	μA
	C_L	Capacitive load				15	pF
INTØ/-INT7/ INTERRUPTS (One line only, switch selected)	V_{OL}	Output low voltage	I_{OL} = 48 mA			0.4	V
	I_{OH}	Output leakage high	Open col output is off V_{OH} = 5.5 V			100	μA
	C_L	Capacitive load				15	pF
DATØ/-DATF/	V_{OL}	Output low voltage	I_{OL} = 15 mA			.45	V
	V_{OH}	Output high voltage	I_{OH} = -1 mA	2.4			V
	V_{IL}	Input low voltage				.85	V
	V_{IH}	Input high voltage		2.0			V
	C_L	Capacitive load				15	pF
	I_{IL}	Input current at V_{IL}	V_{IN} = 0.5 V			0.65	mA
	I_{IH}	Input current at V_{IH}	V_{IN} = 2.4 V			220	μA
INIT/	V_{IL}	Input low voltage				0.8	V
	V_{IH}	Input high voltage		2			V
	I_{IL}	Input current at V_{IL}	V_{IN} = 0.5 V			-12.8	mA
	I_{IH}	Input current at V_{IH}	V_{IN} = 2.7 V			410	μA
	C_L	Capacitive load				50	pF
XACK/,BUSY/	V_{OL}	Output low voltage	I_{OL} = 16 mA			0.4	V
	V_{OH}	Output high voltage	I_{OH} = -2 mA	2.4			V
	V_{IL}	Input low voltage				0.8	V
	V_{IH}	Input high voltage		2			V
	I_{IL}	Input current at V_{IL}	V_{IL} = 0.4 V			-2.04	mA
	I_{IH}	Input current at V_{IH}	V_{IH} = 2.4 V			90	μA
	C_L	Capacitive load				15	pF
+5 volts	I_{+5V}	+5 volts supply current	Worst Case Component Analysis			3.35	A

Table 8-9

INTELLEC® BUS DMA DC CHARACTERISTICS (continued)

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST DESCRIPTION	PARAMETER			
				MIN	TYP	MAX	UNITS
ADRØ/-ADR7/	I_{IL}	Input current at V_{IL}	V_{IL} = 0.45 V			-3.2	mA
	I_{IH}	Input current at V_{IH}	V_{IH} = 2.4 V			120	μA
	V_{IL}	Input low voltage	T_A = 25°C			0.8	V
	V_{IH}	Input high voltage		2.0			V
ADR8/-ADRF/	I_{LH}	Output leakage high	HIGH Z V_0 = 2.4 V			40	μA
	I_{LL}	Output leakage low	HIGH Z V_0 = 0.4 V			-40	μA
BPRN/	V_{IL}					0.8	V[
	V_{IH}			2			V
	I_{IL}		V_{IL} = 0.4			-8.6	mA
	I_{IH}		V_{IH} = 2.4 V			170	μA
	C_L					15	pF
BPRO/,BREQ/	V_{OL}		I_{OL} = 18 mA			0.5	V
	V_{OH}		I_{OH} = -1 mA	2.7			V
	C_L					15	pF
CCLK/,BCLK/	V_{IL}					0.8	V
	V_{IH}			2			V
	I_{IL}					-2	mA
	I_{IH}					50	μA
	C_L					15	pF
INH1/	V_{IL}					0.85	V
	V_{IH}			2.0			V
	I_{IL}		V_{IL} = 0.45 V			-0.25	mA
	I_{IH}		V_{IH} = 5.25 V			10	μA
	C_L					15	pF

NOTE: Test conditions include loading by the DMA board itself for bidirectional signals.

NOTE: V_{CC} = MIN, T_A = 0° to 70°C unless otherwise stated.

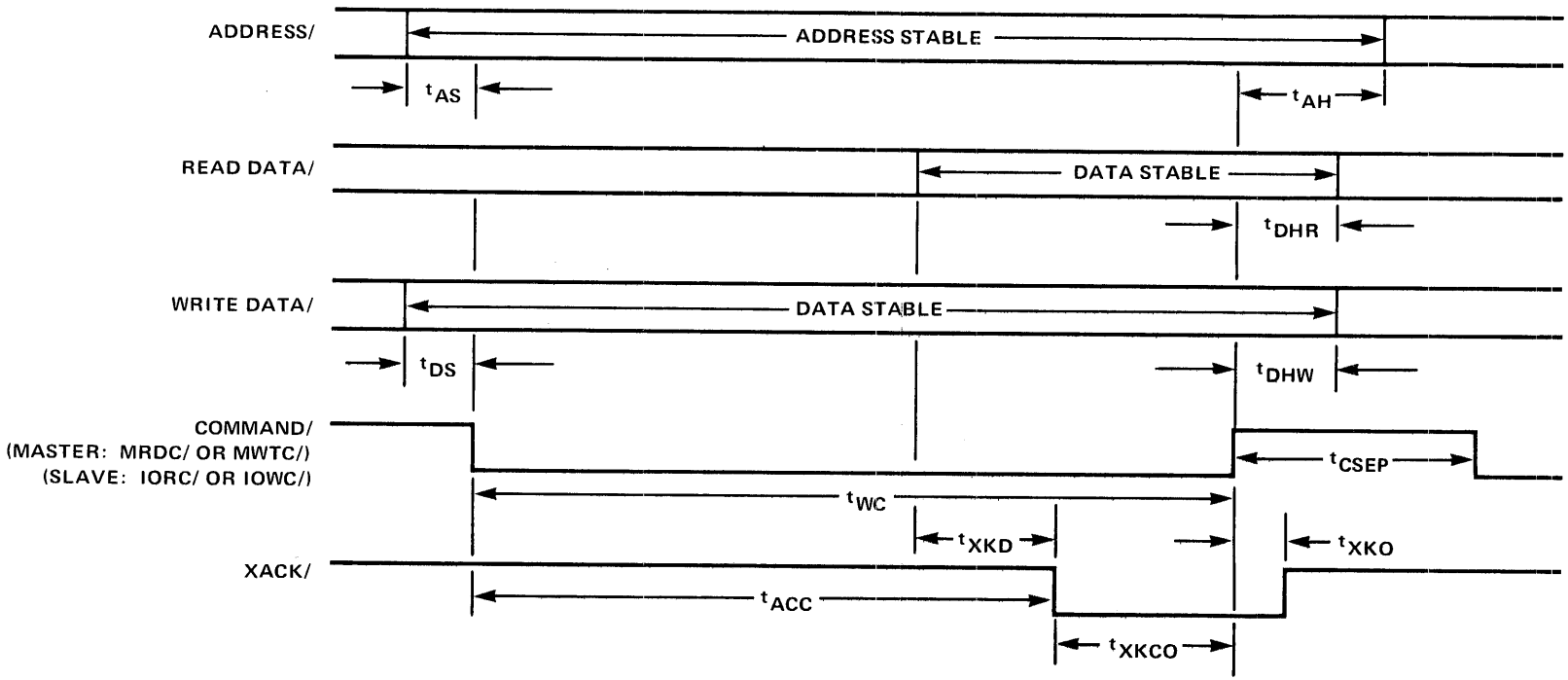


Figure 8-11. Command Timing – Master: Continuous Bus Control

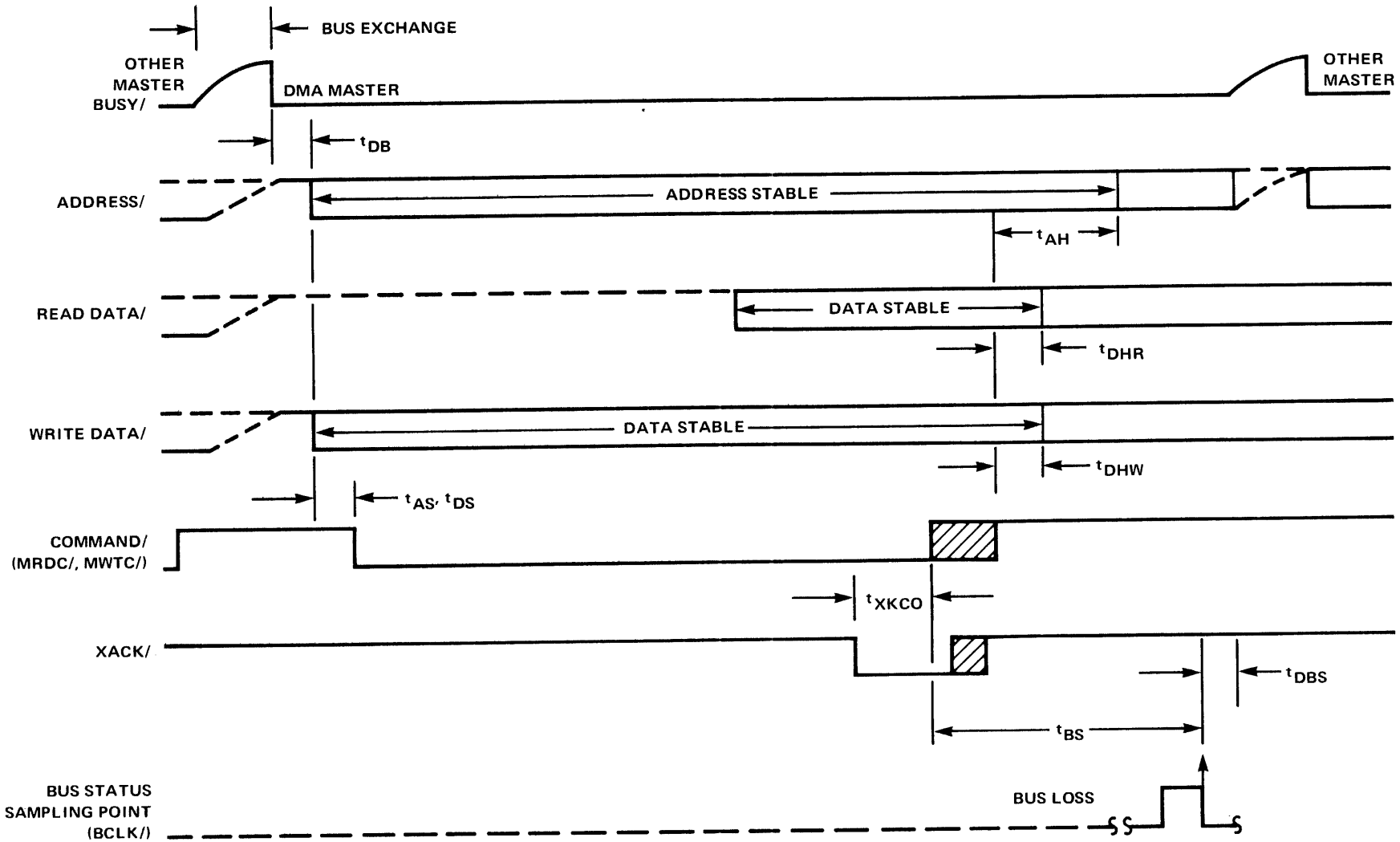


Figure 8-12. Master Command Timing – Bus Exchange

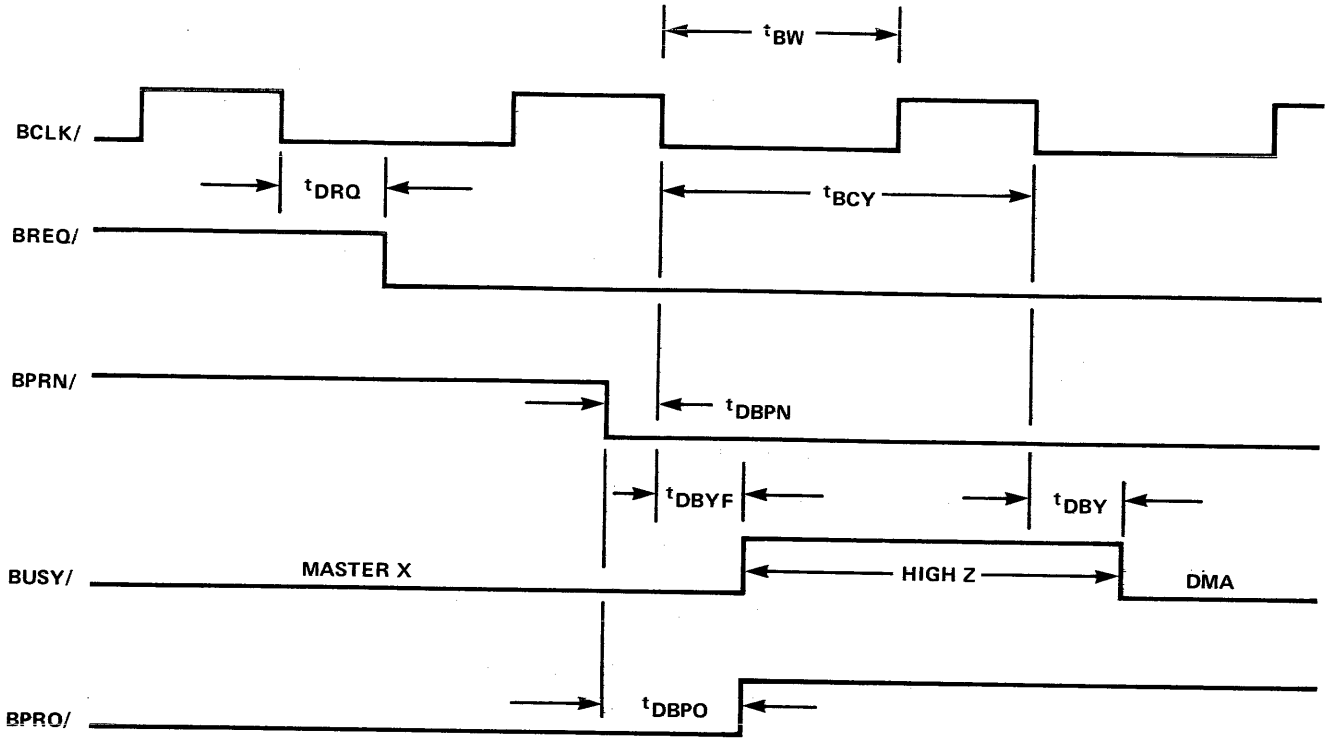


Figure 8-13. Bus Exchange Timing

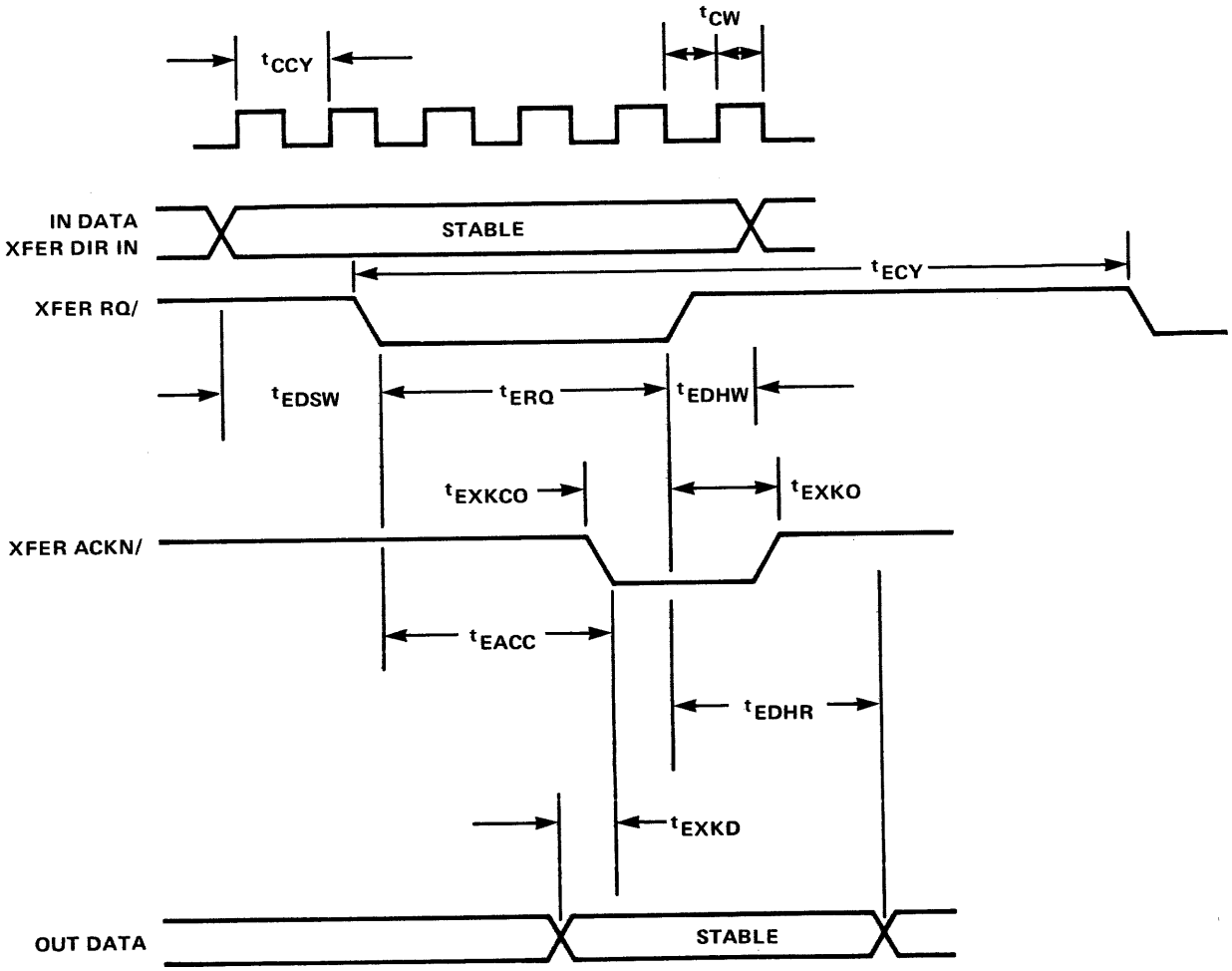


Figure 8-14a. External Transfer Timing

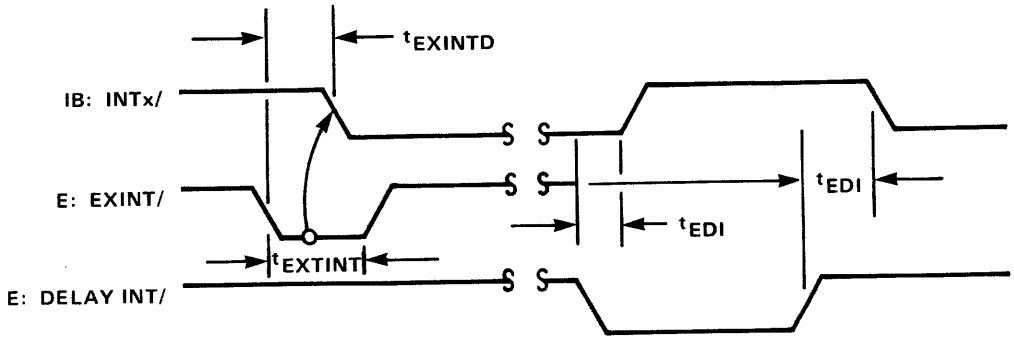


Figure 8-14b. Interrupt Timing

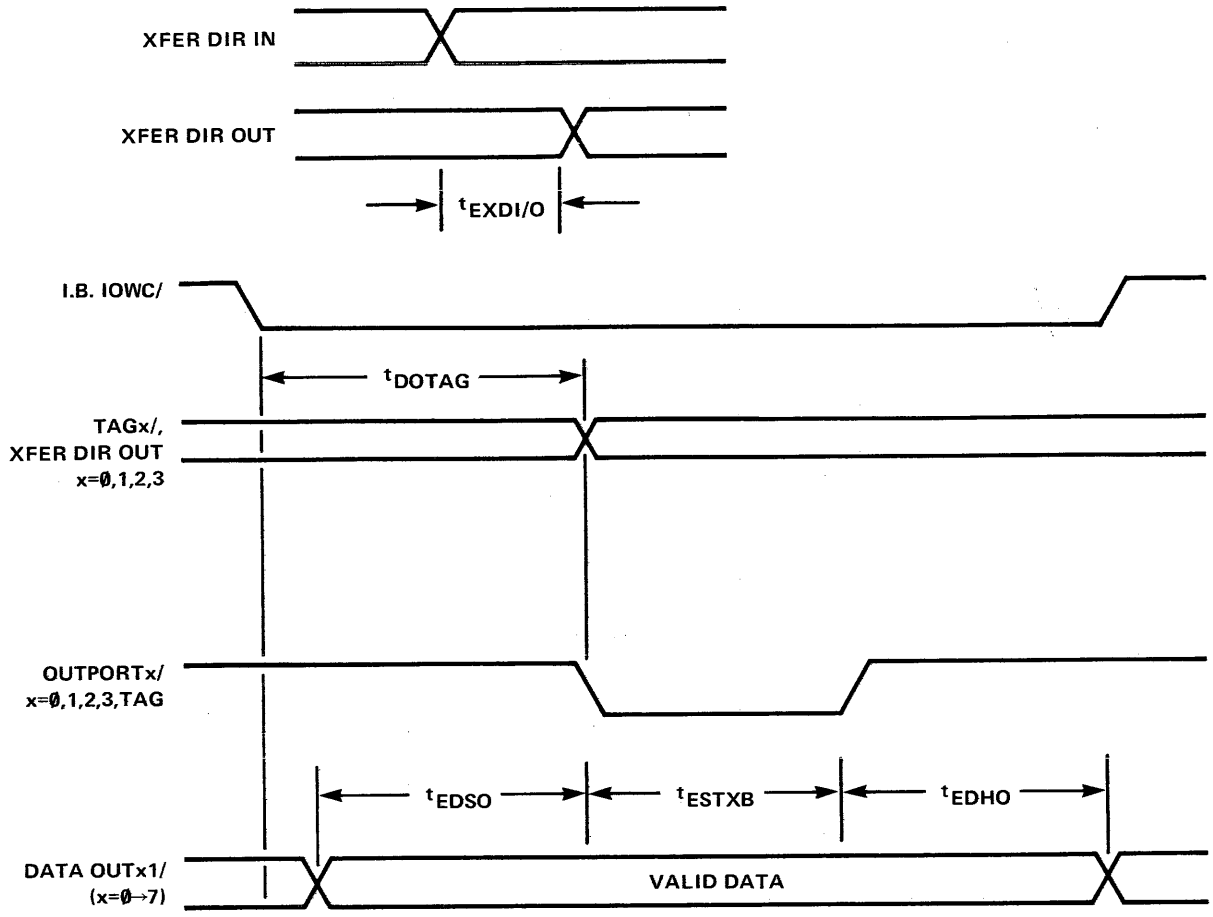


Figure 4-7. Failsafe Logic Timing

Figure 8-15. External I/O Write Timing

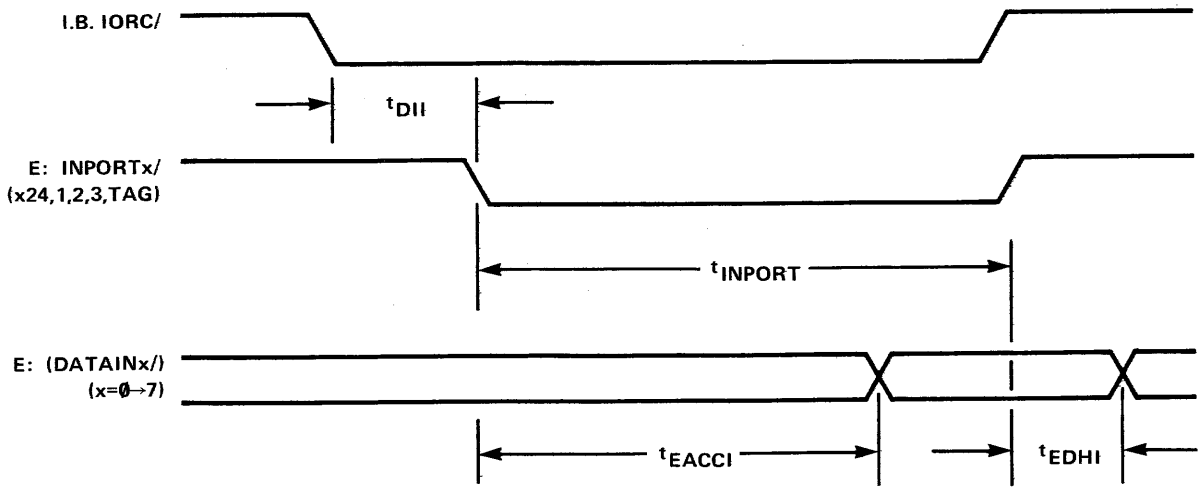


Figure 8-16. External I/O Read Timing

Table 8-10

EXTERNAL INTERFACE DMA DC CHARACTERISTICS

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	PARAMETER		
				MIN	MAX	UNITS
XFER RQ/, XFER DIR IN/, EX INTERRUPT/, STATUSx/ (x=0,1,2,3), DATA IN y/ (y=0→7)	V_{IL}	Input Low Voltage			0.4	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at V_{IL}	V_{IL} = 0.5 V		–44.0	mA
	I_{IH}	Input Current at V_{IH}	V_{IH} = 2.7 V		190	μA
	C_L	Capacitive Load			15	pF
DELAY INT/	V_{IL}				0.4	V
	V_{IH}			2.0		V
	I_{IL}		V_{IL} = 0.5 V		–4.0	mA
	I_{IH}		V_{IH} = 2.4 V		40	μA
	C_L				15	pF
XFER ACKNOWLEDGE/ XFER DIR OUT, TAGx (x=0,1,2,3), OUTPUTy (y=0,1,2,3,TAG), INPUT y, DATA OUT z (z=0→7)	V_{OL}	Output Low Voltage	I_{OL} = 48 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = –1.2 mA	2.4		V
	C_L				15	pF
RESET INTERRUPT	V_{OL}	Output Low Voltage	I_{OL} = 36 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = –1.2 mA	2.4		V
	C_L				15	pF

Chapter 9
INPUT/OUTPUT (I/O) MODULE

The Input/Output (I/O) Module has been designed specifically as an optional component of the INTELLEC MDS System. The general-purpose design of the input and output ports, however, allows the I/O Module to be used in many different types of 8-bit computer systems. Consequently, the I/O Module, like all of the INTELLEC MDS modules, is available independently on an OEM basis.

The I/O Module includes four input and four output ports. Each output port latches 8-bit data words and issues a framed strobe pulse, of selectable duration, to the device. All outputs are driven by TTL level buffer drivers. Each input port also supports 8 bits of data, latched or unlatched. All inputs are terminated by dual-in-line, socket-mounted resistor packs.

The I/O Module includes provisions for accepting eight external interrupt requests, buffering them and driving them on eight interrupt priority level lines. In addition, each of the eight I/O ports includes an interrupt request line that is activated by a strobe pulse from the device, then automatically cleared after the port is serviced. These port interrupt requests can be asserted on the system interrupt priority lines or can be used to feed an interrupt status port on this or another module.

The I/O Module accepts eight address inputs from the CPU. The two least significant address bits select one of the four input or output ports, while the six high-order address bits select the I/O Module. That is, these six high-order bits specify the BASE address of the I/O Module to be accessed. The user can select any one of 64 unique values for the BASE address (switch-selectable).

The I/O Module is implemented on a single, 12-in. X 6.75-in. printed circuit board. The module requires only +5 VDC power. Power and all system signals enter the module through an 86-pin, double-sided edge connector. An auxiliary 60-pin connector is available for use at the designer's discretion, or as a means of reaching various test points. The module communicates with all peripheral devices via a 100-pin, double-sided edge connector, located on the top of the module, opposite the 86 and 60-pin connectors.

9.1 FUNCTIONAL DESCRIPTION OF THE I/O MODULE

The I/O Module can be divided into the following functional blocks:

- I/O address decode block
- Timing control block
- Input ports
- Output ports;

as shown in Figure 9-1.

The *I/O address decode block* determines when a particular I/O Module is being addressed, and then selects one of the four input or output ports to be accessed. An I/O Module is selected by the six most significant I/O address bits (the BASE address). The two least significant bits identify one of the four input or output ports. The BASE address is defined by positioning two nine-position rotary switches in the decode block. The six BASE address bits can provide 64 unique I/O Module select codes.

The *timing control block* provides user-selectable timing for the generation of the transfer acknowledge (XACK/) and output strobe (STBn/) signals. XACK/ acknowledges all input or output operations performed by the I/O Module. STBn/ (where n specifies one of the four ports, 0–3) strobes data, output through the I/O Module, into the proper external device. Timing for XACK/ and STBn/ is selected by connectng one of five jumper pairs in the timing control block.

The four *input ports* and four *output ports* provide a three-state buffered data path between the

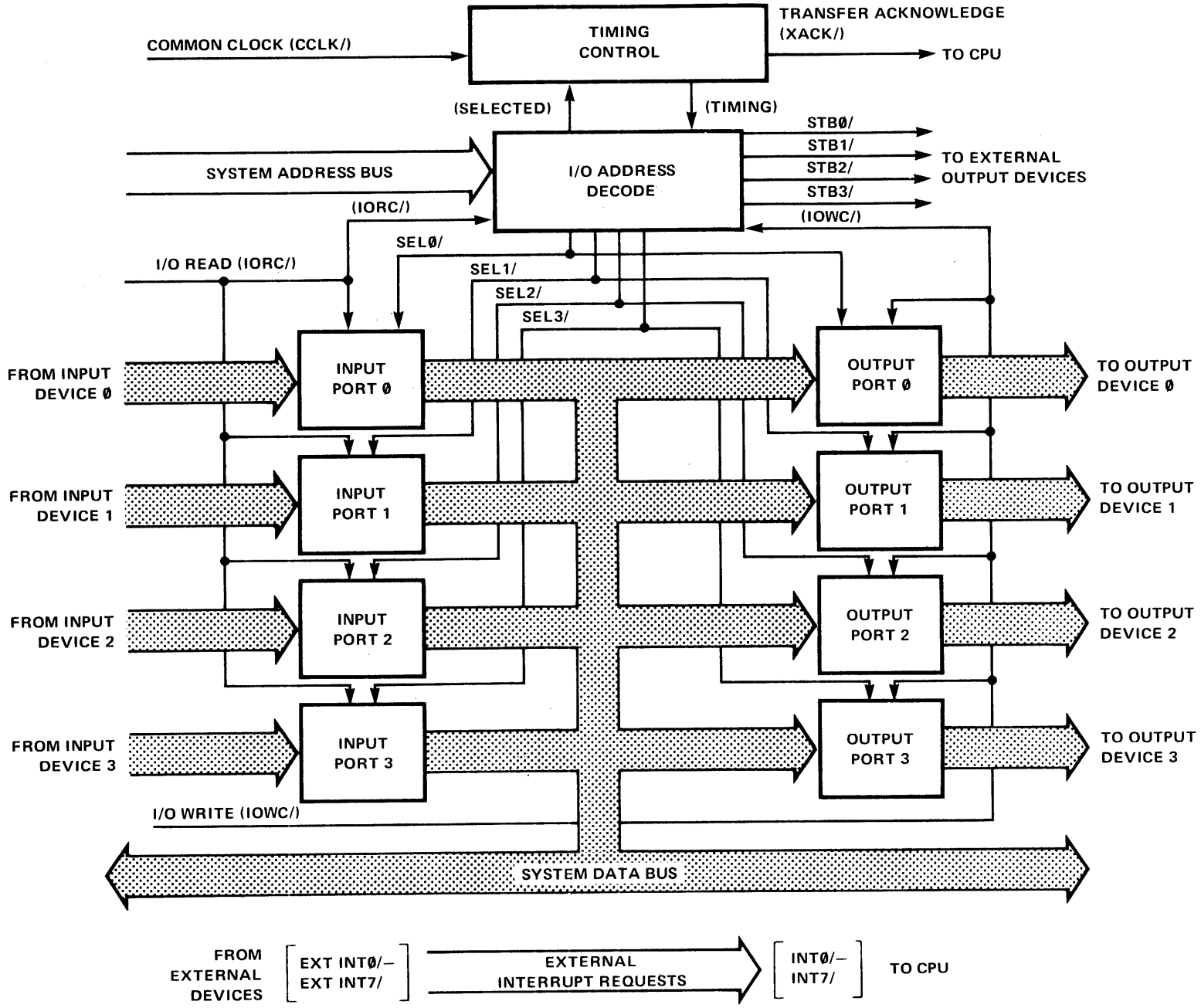


Figure 9-1. I/O Module Functional Block Diagram

system data bus and external devices. Each port is implemented with one of Intel's 8212 8-bit I/O port devices. Data can be latched or unlatched, and interrupt requests can be automatically set and reset by each 8212 device. In addition, each of the eight system interrupt lines can be driven by an external interrupt request line that can be activated by an external device.

9.2 I/O MODULE: THEORY OF OPERATION

The following sub-sections provide a complete description of the theory of operation for each of the functional units on the I/O Module.

The I/O Module accepts/transmits signals, data and power through three different PC edge connectors:

- J1 Peripheral connector (to/from I/O peripherals)
- P1 Bus connector (to/from the system bus)
- P2 Auxiliary connector (to/from the auxiliary bus)

To avoid any ambiguity when referring to connector pins in subsequent paragraphs, we will always list the connector as well as the pin whenever such references are required; for example, P1-14 refers to pin 14 on connector P1. Pin lists for the three connectors are provided in Section 9.3.2.

The schematic (2 sheets) for the I/O Module is provided in Figure 9-6, located in Section 9.2.5.

9.2.1 I/O ADDRESS DECODE BLOCK

The I/O address decode block consists of three 3205 three-to-eight decoders, two nine-position rotary switches (for BASE address selection) and assorted gating circuits as shown on sheet 1 of the module schematic, Figure 9-6.

The two least significant address bits (ADRØ/–ADR1/) feed the two least significant inputs (AØ and A1) on the first 3205 decoder (A14). This decoder is enabled by the outputs of the two rotary switches when the proper BASE address is recognized. The four least significant inverted outputs from this decoder define four select signals (SELØ/, SEL1/, SEL2/, SEL3/). Only one select signal will be true during an I/O cycle. Each SELn/ selects one of the four input and four output ports. The I/O read (IORC/) or write (IOWC/) command determines whether it is input port n or output port n which is selected by SELn/ (see Sections 9.2.2 and 9.2.3).

Each of the four outputs from the decoder also feed four 7402 negative-input AND gates. During output operations (i.e., when IOWC/ is true), the 7402 gate associated with the active SELn/ line is activated for a period determined by the timing control logic (see Section 9.2.2). The 7402 gates feed 7437 NAND gates which drive the output strobes (STBØ/, STB1/, STB2/ and STB3/). STBn/ strobes the output data byte into the device interfaced to port n (see Section 9.2.2 for selectable timing of STBn/ signals).

The six most significant bits (ADR2/–ADR7/) of the 8-bit I/O address specify the BASE address of the I/O Module to be selected. Address lines ADR5/–ADR7/ are applied to the three address inputs of the second decoder (A24), while address lines ADR2/–ADR4/ are applied to the address inputs of the third decoder (A12). Both decoders are permanently enabled. The eight inverted outputs from each decoder feed one position on a nine-position rotary switch.

When ADR5/–ADR7/ specify a binary value equal to the setting of the X1 switch (S1), the output from S1 will be true. Similarly, when ADR2/–ADR4/ specify a binary value equal to the setting of the X2 switch (S2), the output from S2 will be true. The output from S1 and S2 must both be true to enable the port select decoder (A14), as previously mentioned. Note that setting either switch to position 9 disables the I/O Module. Table 9-1 correlates all possible combinations of switch settings with the 64 possible BASE addresses for the I/O Module.

9.2.2 TIMING

The timing control block consists of a 7474 D-type flip-flop, a 74161 synchronous 4-bit counter, a 7493 4-bit binary counter, a five-pair jumper pad for timing selection and various gating circuits, as

Table 9-1

BASE ADDRESS SELECTION

BASE ADDRESS (HEX)	X1 SWITCH (S1)* SETTING	X2 SWITCH (S2)* SETTING	BASE ADDRESS (HEX)	X1 SWITCH (S1)* SETTING	X2 SWITCH (S2)* SETTING
00	1	1	80	5	1
04	1	2	84	5	2
08	1	3	88	5	3
0C	1	4	8C	5	4
10	1	5	90	5	5
14	1	6	94	5	6
18	1	7	98	5	7
1C	1	8	9C	5	8
20	2	1	A0	6	1
24	2	2	A4	6	2
28	2	3	A8	6	3
2C	2	4	AC	6	4
30	2	5	B0	6	5
34	2	6	B4	6	6
38	2	7	B8	6	7
3C	2	8	BC	6	8
40	3	1	C0	7	1
44	3	2	C4	7	2
48	3	3	C8	7	3
4C	3	4	CC	7	4
50	3	5	D0	7	5
54	3	6	D4	7	6
58	3	7	D8	7	7
5C	3	8	DC	7	8
60	4	1	E0	8	1
64	4	2	E4	8	2
68	4	3	E8	8	3
6C	4	4	EC	8	4
70	4	5	F0	8	5
74	4	6	F4	8	6
78	4	7	F8	8	7
7C	4	8	FC	8	8

*Position 9 disables the I/O Module.

shown on sheet 1 of the module schematic, Figure 9-6.

The common clock pulse, CCLK/ (9.8304 MHz), is inverted and applied to the A input of the 7493 counter. The QA output feeds the B input. Consequently, the QA output divides CCLK by 2, QB divides CCLK by 4, QC divides CCLK by 8 and QD divides CCLK by 16. That is, CCLK defines a 100-ns (approximately) period pulse, QA defines a 200-ns pulse, QB a 400-ns pulse, QC an 800-ns pulse and QD a 1600-ns pulse. Any one of these timing signals can be selected to drive the timing control logic by connecting the proper jumper pair, as listed in Table 9-2.

The selected timing pulse (t_1) feeds the clock input on the 74161 synchronous counter. This counter is cleared by the absence of either an I/O read (IORC/) or write (IOWC/) command. When IORC/ or IOWC/ goes true, however, the counter begins counting from zero. The QA output goes high with the first timing pulse and every other alternate pulse. When the QB output from the 74161 counter goes high (minimum=t_1, maximum=$2t_1$, after IORC/ or IOWC/ goes true), one of the output

Table 9-2

TIMING SELECTION

TIMING PULSE (t_1) (NANOSECONDS, NOMINAL)	JUMPER CONNECTION
100	1–2
200	3–4
400	5–6
800	7–8
1600	9–10

(Also refer to Figure 9-2)

strobe signals (STBn/) is enabled if IOWC/ is also true (i.e., if it is an output instruction), as mentioned in Section 9.2.1. QB remains high for two timing pulses ($2t_1$). QC goes high as QB goes low. One t_1 period later, QA goes high and clocks the high level on QC into the 7474 latch. The Q output from the 7474 section causes the 74161 counter to be cleared. The $\overline{Q}$ output feeds a 74125 circuit which drives XACK/ (via pin P1-23). XACK/ is enabled until IORC/ or IOWC/ goes false.

Figure 9-2 illustrates timing for the STBn/ and XACK/ signals.

9.2.3 INPUT OPERATIONS

The I/O Module includes four input ports, each implemented with an Intel 8212 device, as shown on sheet 2 of the module schematic, Figure 9-6. Figure 9-3 illustrates the logic within an 8212 device. In the input mode the MD input is held low (grounded).

If the data input by an external device is to be latched in the 8212 I/O port, or if an interrupt is to be generated when the data is input to the 8212, the device will accompany the input data byte with

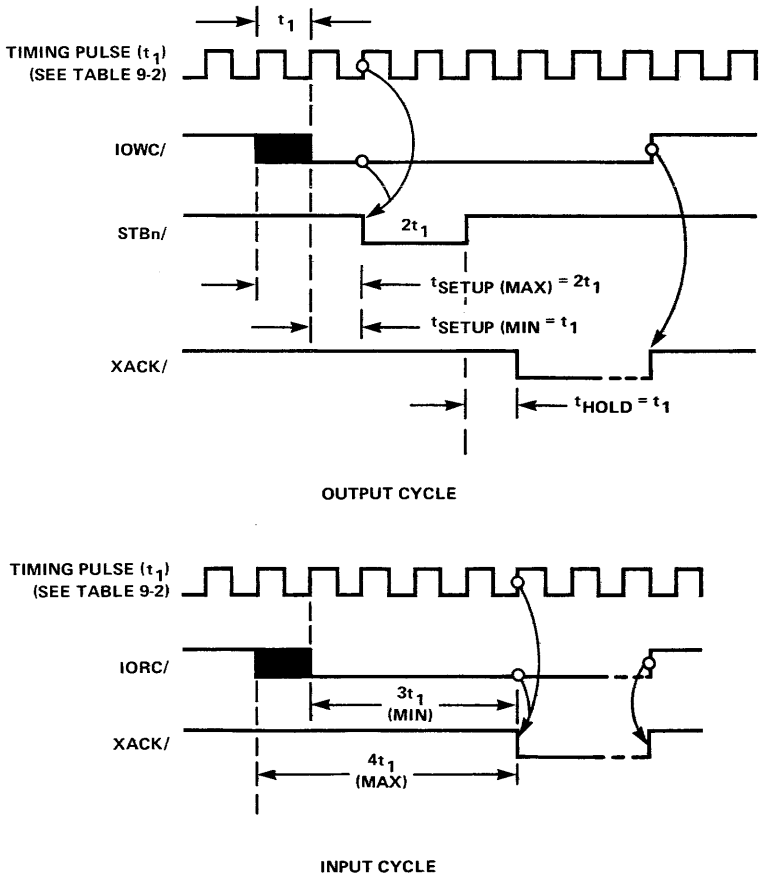


Figure 9-2. I/O Module Timing

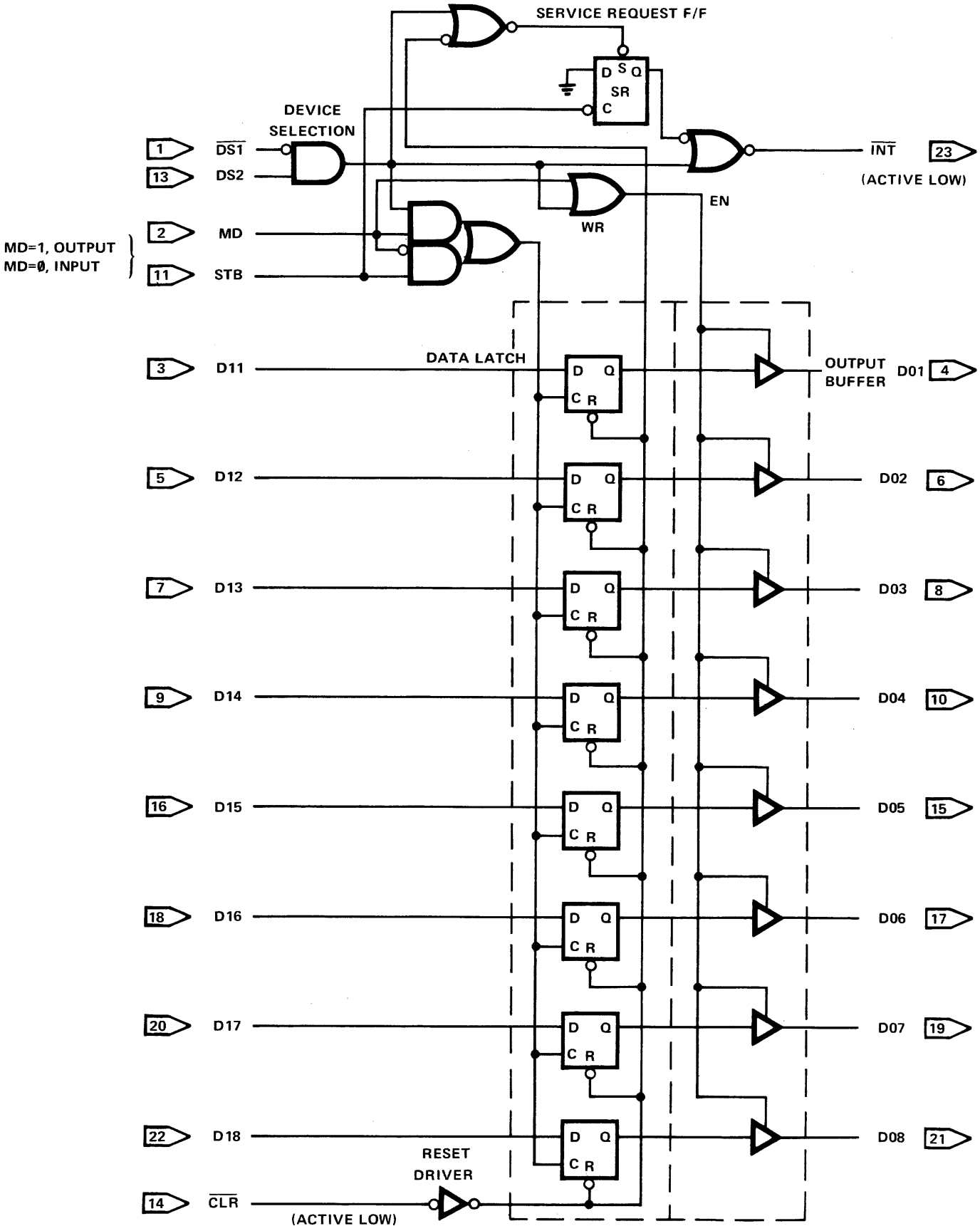


Figure 9-3. 8212 Logic Diagram

a strobe signal (ISTBn, where n is the port number). The data lines from the device are terminated by resistors provided by dual-in-line, socket-mounted resistor packs, and applied to the 8212. If ISTBn is present, the data will be latched in the 8212 when ISTBn goes low; the data will remain latched until ISTBn goes high again. If the strobe is not present, the output of a latch will always reflect the level on its input. In such a case, the data levels must be maintained by the device.

The negative-going edge of the strobe also clocks the service request flip-flop in the 8212, causing the interrupt request line (IINTn/) to go true (low). IINTn/ can be asserted on a system interrupt line or can be connected to one bit of an interrupt status register.

NOTE: Even if a strobe is not present, IINTn/ will be generated when the port is selected; consequently, IINTn/ should not be connected to the system interrupt input if interrupts are not desired.

When the CPU executes an I/O read instruction to a particular port, a select signal (SELn/) from the I/O address decode block enables the appropriate port as long as IORC is true. IORC (DS2) and SELn/ (DS1) enable the eight three-state output buffers within the 8212. These buffers drive the data on the system data bus (DAT0/–DAT7/).

The system reset signal (SYS RST) clears the latches and service request flip-flops in all of the 8212 I/O ports.

Figure 9-4 illustrates the timing within an input port.

In addition to the IINTn/ line from each 8212, an external device can always request an interrupt by pulling one of the eight external interrupt lines (EXINTm/) low (see sheet 1 of the module schematic), activating one of the INTm/ lines that are driven by 7407 non-inverting, open-collector buffers.

9.2.4 OUTPUT OPERATIONS

The I/O Module includes four output ports, each implemented with an Intel 8212 device, as shown on sheet 2 of the module schematic, Figure 9-6. Refer to Figure 9-3 for a logic diagram of the 8212. In the output mode, the MD input is held high.

When a peripheral device is ready to accept a data byte, it can request an interrupt by issuing an output strobe signal (OSTBn). OSTBn, *from* a device, is not to be confused with the output strobe signal (STBn/) that is sent *to* a device during an output cycle.

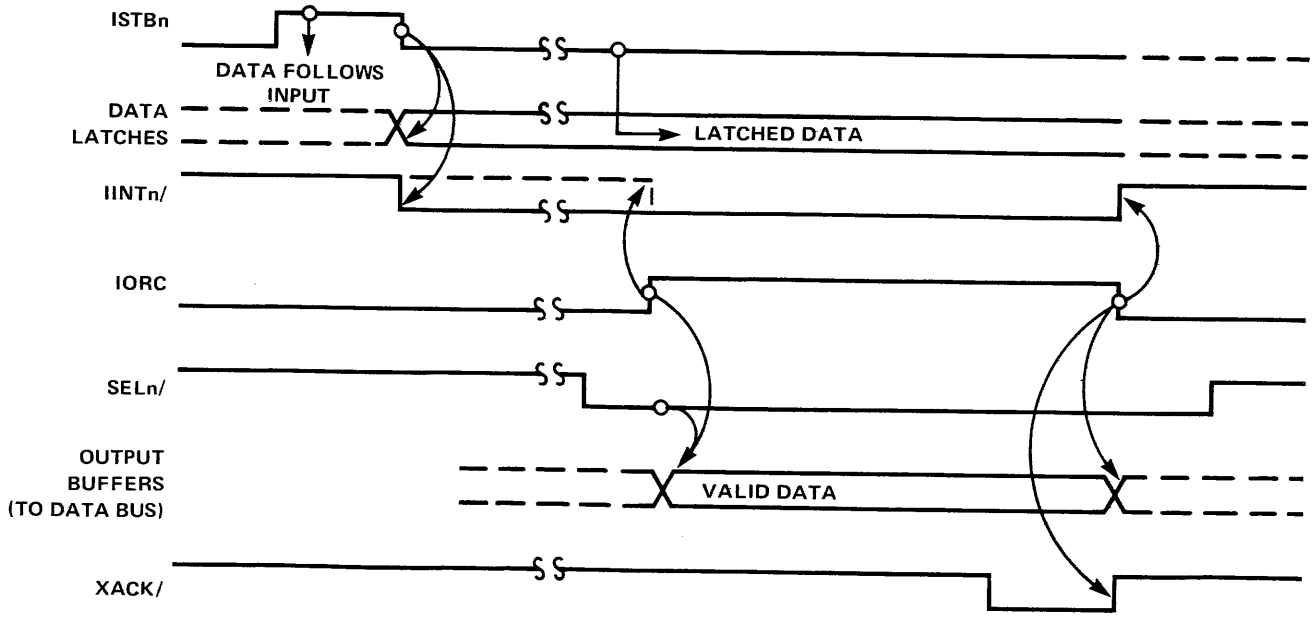


Figure 9-4. Input Port Timing

The service request flip-flop in the 8212 will be clocked on the negative-going edge of OSTBn, causing the interrupt request signal for that device, OINTn/, to go true (low). The device can also request an interrupt by pulling one of the eight external interrupt lines (EXINTm/) low. The primary difference between the two types of request is that the 8212's interrupt line (OINTn/) is automatically cleared after the port is serviced, whereas the device is responsible for clearing its external interrupt line (EXINTm/).

NOTE: OINTn/, like IINTn/ on input ports, is generated when the port is selected, even if a strobe (OSTBn) was not issued by the device.

When the CPU executes an I/O write instruction to a particular port, a select signal (SELn/) from the I/O address decode block enables the appropriate port as long as IOWC is true. The output from the port latches will reflect the levels on the inputs (from the system data bus) until IOWC or SELn go false, at which time the data is latched. The 8212 output buffers are always enabled in the output mode (i.e., when MD is high). The 7437 buffers drive data at active-low, TTL levels.

The system reset signal (SYS RST) clears the latches and service request flip-flops in all of the 8212 I/O ports. SYS RST/ is also available to the external devices (pin J1-54) via a 74125 non-inverting buffer.

Figure 9-5 illustrates the timing within an output port.

9.2.5 I/O MODULE SCHEMATIC

Figure 9-6 provides a complete schematic drawing (2 sheets) of all logic on the I/O module.

9.3 UTILIZATION: I/O MODULE

This section provides information on utilization of the I/O Module.

9.3.1 INSTALLATION

In installing the I/O Module, the user must take account of:

(a) environmental extremes
(b) mounting considerations
(c) electrical connections

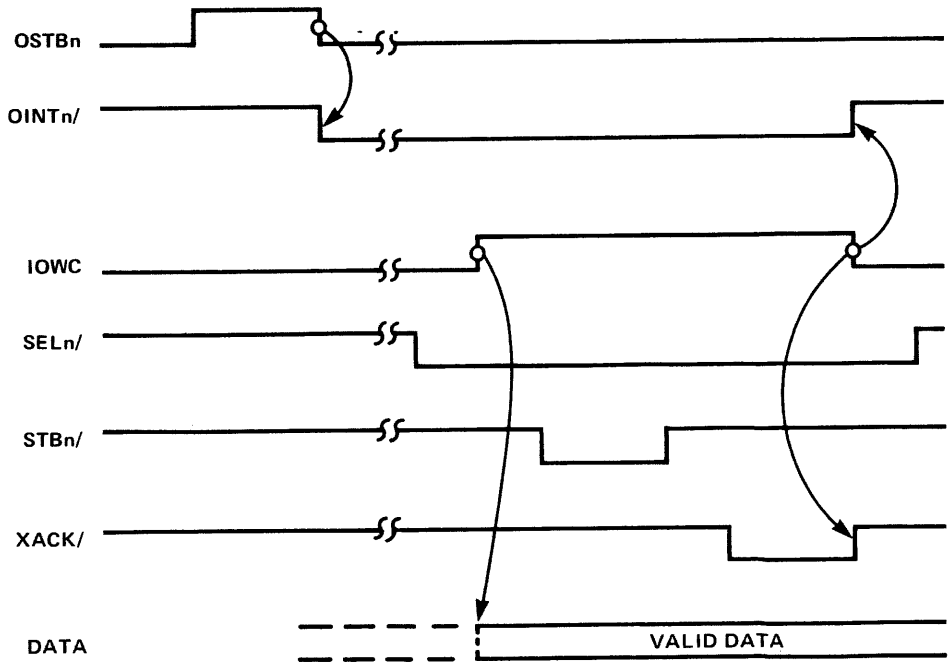
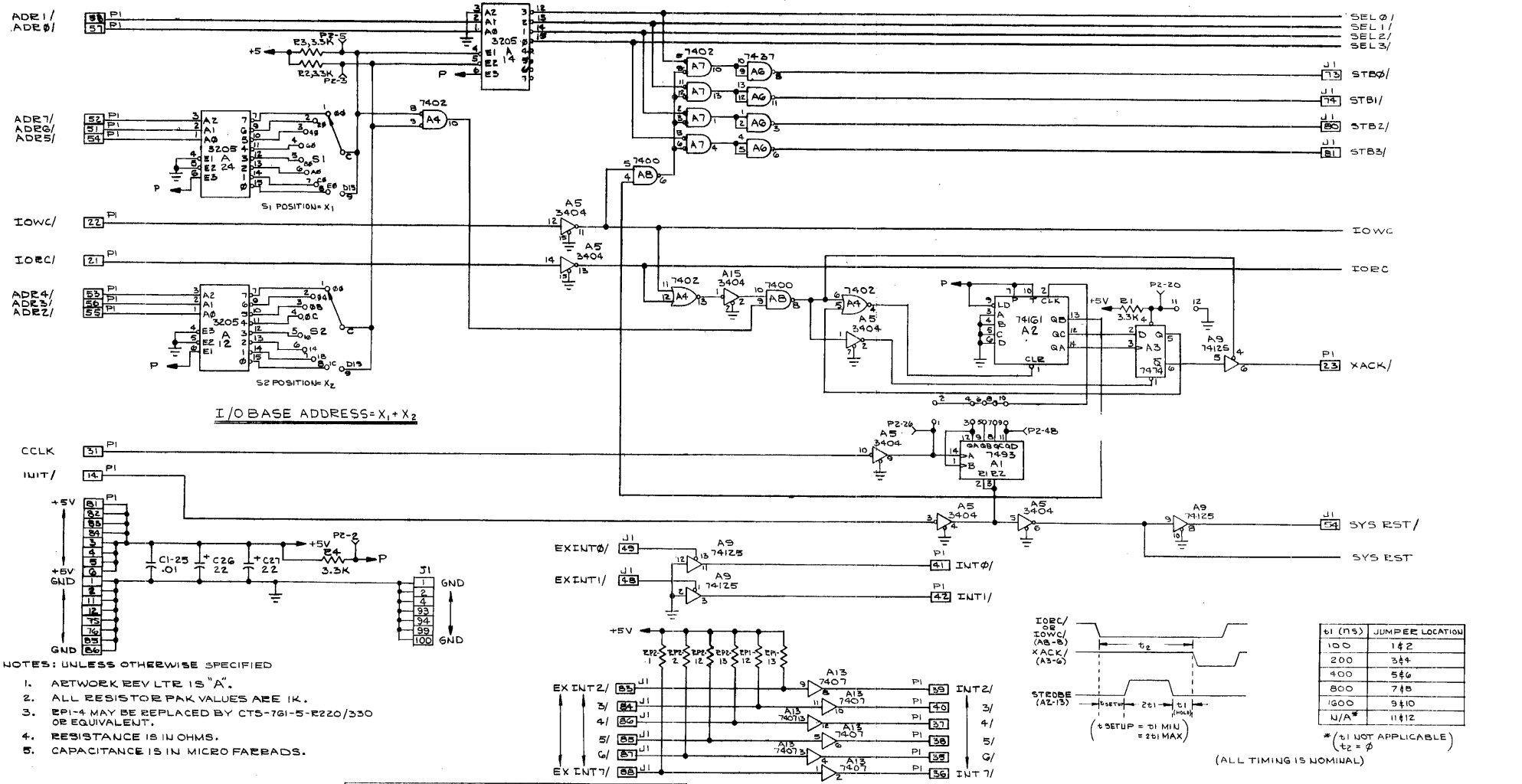


Figure 9-5. Output Port Timing



t1 (ns)	JUMPER LOCATION
100	1&2
200	3&4
400	5&6
800	7&8
1600	9&10
N/A*	11&12

*(t1 NOT APPLICABLE t2 = 0)

SPARES			REF DES	
REF DES	DEVICE TYPE	QTY	LAST USED	NOT USED
A8	7400	2	A33	
A4	7402	1	RP4	
A3	7474	1	R4	
A15	3404	3	C27	
			SW2	

intel © INTEL 1975	3065 BOWERS AVE. SANTA CLARA CALIF. 95051	
TITLE: SCHEMATIC, I/O MODULE		
SIZE: D	DEPT: 410	DRAWING NO.: 2000361 REV: D

Figure 9-6. I/O Module Schematic (Sheet 1 of 2)

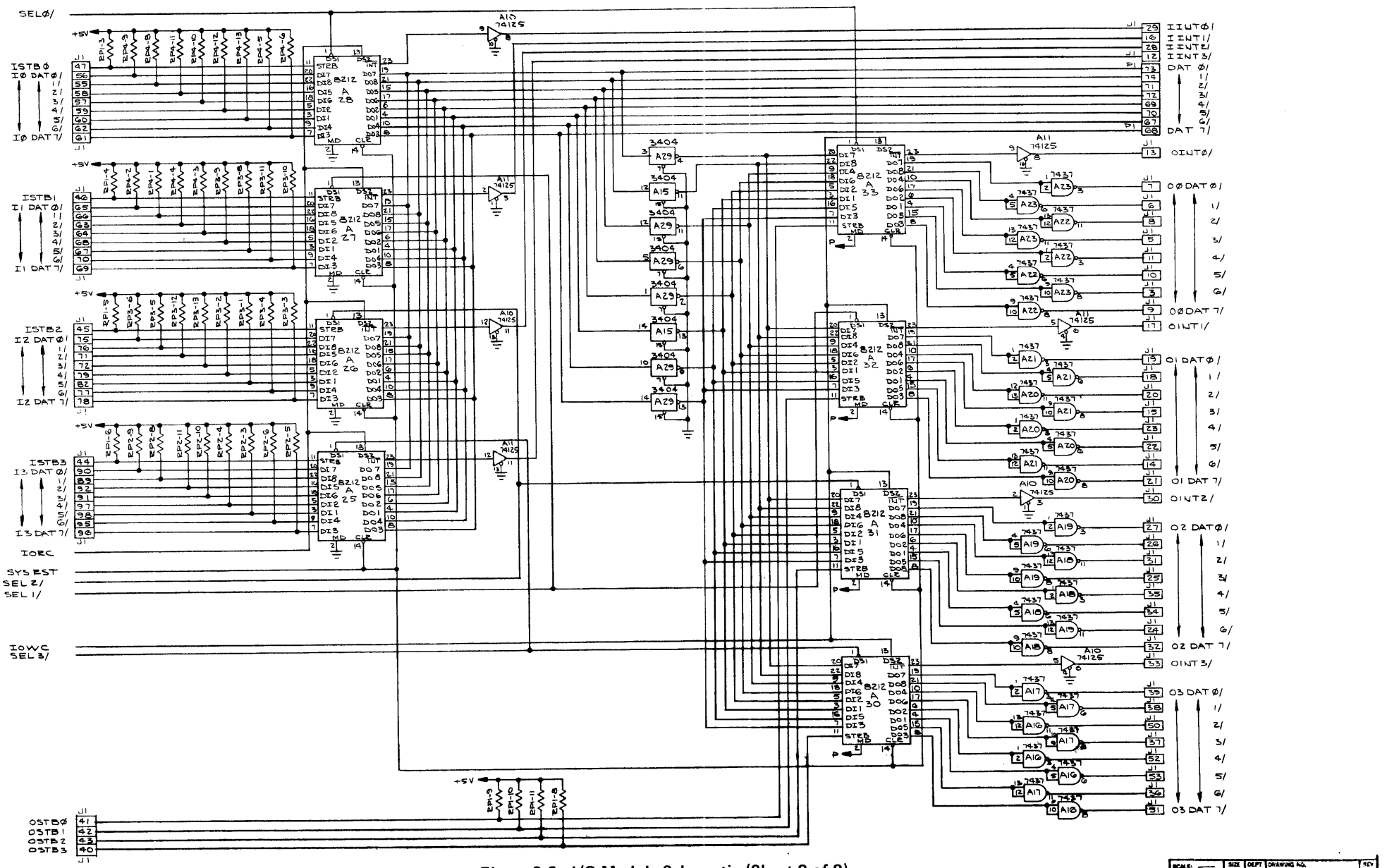


Figure 9-6. I/O Module Schematic (Sheet 2 of 2)

(d) power requirements
(e) signal requirements
(f) base address selection
(g) timing selection
(h) bus termination packs

Environment

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conducive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity should not exceed 90%, non-condensing.

Mounting

Avoid locating the module near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections on the board, resulting in abnormally high noise levels or outright failure of the assembly.

Dimensions of the module are 12-in. X 6.75-in. Be sure to allow enough additional clearance to ensure adequate cooling.

The module is designed to plug directly into three standard, double-sided PC edge connectors. An 86-pin connector and a 60-pin auxiliary connector are located on one edge of the board; a 100-pin connector is on the opposite edge. The connectors can serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the assembly be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the board, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the components on the module.

Electrical Connections

The I/O Module communicates with the motherboard and, consequently, the rest of the system, through a standard 86-pin, double-sided PC edge connector (P1), 0.156-in. contact centers, as shown in Figure 9-7. Control Data VPB01E43A00A1 is one suitable type of connector. Pin allocations on this connector are given in Table 9-3 of Section 9.3.2. The module can also communicate with other modules in the system, through the auxiliary 60-pin, double-sided PC edge connector (P2), 0.1-in. contact centers (see Figure 9-7). Pin allocations for this connector (primarily test points) are listed in Table 9-4. The module transfers information to/from the peripheral devices via a 100-pin, double-sided PC edge connector (J1) which attaches to the edge opposite that of the other two connectors. This connector has 0.1-in. contact centers. Viking 3VH50/1JN5 is one suitable type of connector for communicating with the peripheral devices. Pin allocations for this connector are given in Table 9-5.

The I/O Module requires only +5 VDC power.

Refer to the pin list in Table 9-3 of Section 9.3.2 for power connections.

Signal Requirements

All data and control functions appearing at the module edge connectors are at TTL levels. Electrical characteristics of the signal inputs and outputs, as well as power inputs are given in Section 9.4.

Signal descriptions and connector pin allocations are given in Section 9.3.2.

Base Address Selection

The six most significant bits of the 8-bit I/O address define the BASE address for an I/O Module. The user must select a BASE address by positioning the X1 and X2 rotary switches as listed in Table 9-1. Notice that the I/O Module is disabled if either switch is in position 9.

Timing Selection

The user must also select timing parameters for the generation of the output strobe (STBn/) and

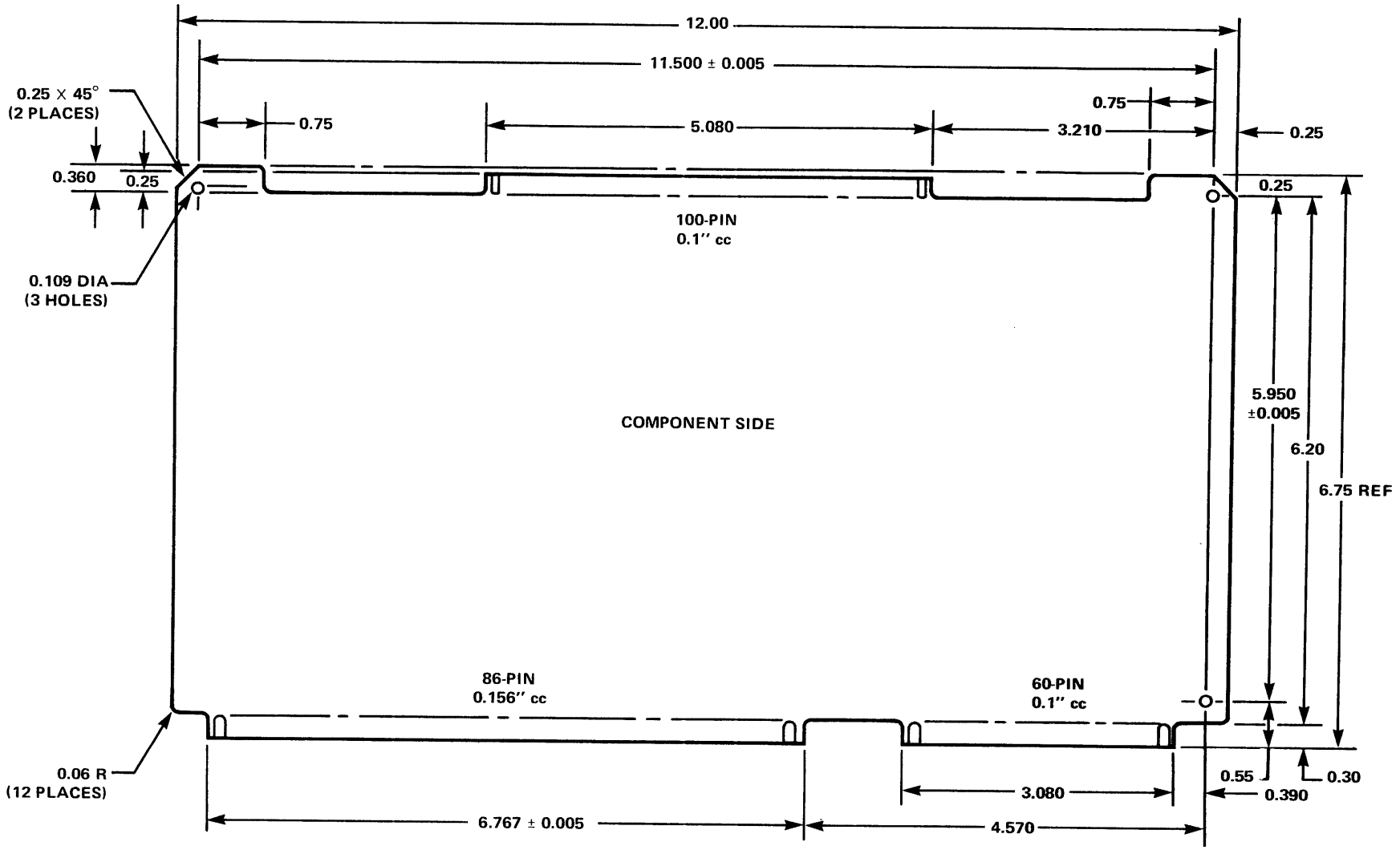


Figure 9-7. I/O Module Connectors

transfer acknowledge (XACK/) signals, that meet the timing requirements of the CPU and/or the external output devices being used. Specific timing is selected by connecting one of the five jumper pairs in the timing control block, as listed in Table 9-2. Figure 9-2 illustrates module timing relative to the timing pulse (t_1) selected.

Bus Termination Packs

The I/O Module includes provisions for installing dual-in-line, socket-mounted resistor packs for the purpose of terminating data lines from the external input devices (refer to sheet 2 of the module schematic). The user should install appropriate packs to match the drive characteristics of the external device with the DC characteristics of the 8212 data input lines.

9.3.2 PIN LISTS: I/O MODULE

The following section provides connector pin allocations on the I/O Module. The pins and their designated signal functions for the 86-pin connector (P1) are listed in Table 9-3. The same information for the 60-pin auxiliary connector (P2) is listed in Table 9-4. Pin and signal information for the 100-pin peripheral connector (J1) is given in Table 9-5.

9.4 OPERATING CHARACTERISTICS: I/O MODULE

The AC and DC characteristics of all major signals that appear at the edge connectors will be listed in this section.

9.4.1 AC CHARACTERISTICS

AC characteristics are listed in Tables 9-6a and 9-6b.

9.4.2 DC CHARACTERISTICS

DC characteristics are listed in Tables 9-7a and 9-7b. Power requirements are cited below:

	TYP	MAX
V_{CC} +5 VDC ±5%	1.9A	2.6A

Table 9-3

P1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	GND	Ground
2	GND	
3	+5 VDC	Power inputs
4	+5 VDC	
5	+5 VDC	
6	+5 VDC	
7		
8		
9		
10		
11	GND	Ground
12	GND	
13		
14	INIT/	System reset
15		
16		
17		
18		
19		
20		
21	IORC/	I/O read command
22	IOWC/	I/O write command
23	XACK/	Acknowledge transfer
24		
25		
26		
27		
28		
29		
30		
31	CCLK/	Common clock (9.8304 MHz)
32		
33		
34		
35	INT6/	Interrupt requests
36	INT7/	
37	INT4/	
38	INT5/	
39	INT2/	
40	INT3/	
41	INT0/	
42	INT1/	
43	ADRE/	Address bus

PIN	SIGNAL	FUNCTION
44	ADRF/	Address bus
45	ADRC/	
46	ADRD/	
47	ADRA/	
48	ADRB/	
49	ADR8/	
50	ADR9/	
51	ADR6/	
52	ADR7/	
53	ADR4/	
54	ADR5/	
55	ADR2/	
56	ADR3/	
57	ADR0/	
58	ADR1/	
59		
60		
61		
62		
63		
64		
65		
66		
67	DAT6/	Data bus
68	DAT7/	
69	DAT4/	
70	DAT5/	
71	DAT2/	
72	DAT3/	
73	DAT0/	
74	DAT1/	
75	GND	Ground
76	GND	
77		
78		
79		
80		
81	+5 VDC	Power inputs
82	+5 VDC	
83	+5 VDC	
84	+5 VDC	
85	GND	Ground
86	GND	

Table 9-4

P2 CONNECTOR PIN LIST TEST POINTS

PIN	SIGNAL	FUNCTION	PIN	SIGNAL	FUNCTION
1			31		
2		Pull-up	32		
3	X2 EN/	X2 Address Select	33		
4			34		
5	X1 EN/	X1 Address Select	35		
6			36		
7			37		
8			38		
9			39		
10			40		
11			41		
12			42		
13			43		
14			44		
15			45		
16			46		
17			47		
18			48		
19			49		
20	SET XACK/	XACK direct set	50		
21			51		
22			52		
23			53		
24			54		
25			55		
26	CCLK/	Common clock	56		
27			57		
28			58		
29			59		
30			60		

Table 9-5

J1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	GND	Ground
2	GND	
3	O0DAT6/	Data out bus from port 0, but 6
4	GND	Ground
5	O0DAT3/	Data out bus from output port 0
6	O0DAT1/	
7	O0DAT0/	
8	O0DAT2/	
9	O0DAT7/	
10	O0DAT5/	
11	O0DAT4/	
12	IINT3/	Interrupt from input port 3
13	OINT0/	Interrupt from output port 0
14	OIDAT6/	Data out bus from output port 1
15	OIDAT3/	
16	IINT1/	Interrupt from input port 1
17	OINT1/	Interrupt from output port 1
18	O1DAT1/	Data out bus from output port 1
19	O1DAT0/	
20	O1DAT2/	
21	O1DAT7/	
22	O1DAT5/	
23	O1DAT4/	
24	O2DAT6/	Data out bus from output port 2
25	O2DAT3/	
26	O2DAT1/	
27	O2DAT0/	
28	IINT2/	Interrupt from input port 2
29	IINT0/	Interrupt from input port 0
30	OINT2/	Interrupt from output port 2
31	O2DAT2/	Data out bus from output port 2
32	O2DAT7/	
33	OINT3/	Interrupt from output port 3
34	O2DAT5/	Data out bus from output port 2
35	O2DAT4/	
36	O3DAT6/	Data out bus from output port 3
37	O3DAT3/	
38	O3DAT1/	
39	O3DAT0/	
40	OSTB3	Output strobes from devices
41	OSTB0	
42	OSTB1	
43	OSTB2	
44	ISTB3	Input strobes from devices
45	ISTB2	
46	ISTB1	
47	ISTB0	
48	EXTINT1/	External interrupt requests
49	EXTINT0/	
50	O3DAT2/	Data out bus from output port 3

Table 9-5

J1 CONNECTOR PIN LIST (continued)

PIN	SIGNAL	FUNCTION
51	O3DAT7/	
52	O3DAT4/	Data out bus from output port 3
53	O3DAT5/	
54	SYS RST/	System reset
55	I0DAT1/	
56	I0DAT0/	
57	I0DAT3/	
58	I0DAT2/	Data in bus to input port 0
59	I0DAT4/	
60	I0DAT5/	
61	I0DAT7/	
62	I0DAT6/	
63	I1DAT2/	
64	I1DAT3/	
65	I1DAT0/	
66	I1DAT1/	Data in bus to input port 1
67	I1DAT5/	
68	I1DAT4/	
69	I1DAT7/	
70	I1DAT6/	
71	I2DAT2/	Data in bus to input port 2
72	I2DAT3/	
73	STB0/	Output strobes to devices
74	STB1/	
75	I2DAT0/	
76	I2DAT1/	
77	I2DAT6/	Data in bus to input port 2
78	I2DAT7/	
79	I2DAT4/	
80	STB2/	Output strobes to devices
81	STB3/	
82	I2DAT5/	Data in bus to input port 2
83	EXT INT2/	
84	EXT INT3/	
85	EXT INT5/	External interrupt requests
86	EXT INT4/	
87	EXT INT6/	
88	EXT INT7/	
89	I3DAT1/	
90	I3DAT0/	Data in bus to input port 3
91	I3DAT3/	
92	I3DAT2/	
93	GND	Ground
94	GND	
95	I3DAT6/	
96	I3DAT7/	Data in bus to input port 3
97	I3DAT4/	
98	I3DAT5/	
99	GND	Ground
100	GND	

Table 9-6a

BUS I/O MODULE (INTELLEC® BUS)

PARAMETER	OVERALL MIN.	OVERALL MAX.	DESCRIPTION INPUT REQUIREMENTS	REMARKS
t_{AS}	49		Address Setup Time To Command	
t_{AH}	49		Address Hold Time From Command	
t_{DS}	44		Data Setup Time To Command, Write	
t_{DHW}	45		Data Hold Time From Command, Write	
t_{SEP}	100		Command Separation	
t_{WC}	t_{ACC}		Command Width	
t_{XKCO}	0		Command Turn Off Delay From SACK/	
t_{BCY}	100		Bus Clock Cycle Time	
t_{BW}	25		Bus Clock Low and High Periods	
t_{CCY}	100		Com. Clock Cycle Time	
t_{CW}	25		Com. Clock Low and High Periods	
OUTPUT LIMITS				
t_{XKO}		80	XACK/ Turn Off Delay	
t_{DHR}	6		Data Hold From Read Command	
t_{ACC}		79	XACK Delay From Command	Jumper Location 11 & 12
		$5nt_{CCY} + 179$	XACK Delay From Command	Jumper Loc's / n / t_{CCY}
		679	XACK Delay From Command	Holes 1–2 / 1 / 100 ns
		1179	XACK Delay From Command	Holes 3–4 / 2 / 100 ns
		1679	XACK Delay From Command	Holes 5–6 / 4 / 100 ns
		2179	XACK Delay From Command	Holes 7–8 / 8 / 100 ns
		2679	XACK Delay From Command	Holes 9–10 / 16 / 100 ns
t_{XKD}	–41		XACK DELAY FROM VAL RD DATA	Jumper Loc's 11 & 12
	$5nt_{CCY} - 33$			SEE t_{ACC}, above

Table 9-6b

I/O MODULE EXTERNAL INTERFACE AC CHARACTERISTICS

PARAMETER	OVERALL		DESCRIPTION OUTPUT LIMITS	REMARKS
	MIN.	MAX.		
t_{EDSO}*	nt_{CCY} –53		OxDATy/setup to STBx/strobe	For n, see Table .a t_{ACC} REMARKS
t_{ESTB}*	$2nt_{CCY}$ –61		STBx/ STROBE WIDTH	
t_{EDHO}*	nt_{CCY} +28		Output Data Hold From STBX/ strobe	
t_{DODAT}		74	IOWC/to valid OxDATy/	
t_{DSTBI}		58	OSTBx/or ISTBx/ Delay to OINTx/or IINTx/	
t_{DINT}		70	IOWC/or IORC/ Delay to OINTx/ or IINTx/	
t_{INTRST}		64	IOWC/or IORC/ Delay to RST I,OINTx/	
t_{EXINT}		30	EXINTx/Delay to INTx/	
INPUT REQUIREMENTS				
t_{STB}	25		OSTBx, ISTBx WIDTH	
t_{EDSSTB}	15		IxDATy/setup to ISTBx/	
t_{EDHSTB}	20		IxDATy/hold from ISTBx/	
t_{EDSR}	0		IxDATy/setup to IORC/	
t_{EDHR}	0		IxDATy/hold from IINTx/	

*Not applicable for jumper position 11-12.

Table 9-7a

INTELLEC® BUS I/O MODULE DC CHARACTERISTICS

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	PARAMETER		
				MIN.	MAX.	UNITS
ADRØ/→ADR7,	V_{IL}	Input Low Voltage			0.85	V
IOWC/, IORC/,	V_{IH}	Input High Voltage		2.0		V
CLLK/, INIT/	I_{IL}	Input Current at V_{IL}	V_{IL} = 0.45 V		–0.25	mA
	I_{IH}	Input Current at V_{IH}	V_{IH} = 5.25 V		10	μA
	C_L	Capacitive Load			15	pF
XACK/	V_{OL}	Output Low Voltage	I_{OL} = 16 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = –5.2 mA	2.4		V
	I_{LH}	Output Leakage High	High Z V_0 = 2.4 V		40	μA
	I_{LL}	Output Leakage Low	High Z V_0 = 0.4 V		–40	μA
	C_L				15	pF
INTØ/→INT7/	V_{OL}		I_{OL} = 16 mA		0.4	V
	I_{OH}		Output is OFF V_{OH} = 5.5 V		250	μA
	C_L				15	pF
DATØ→DAT7/	V_{OL}		I_{OL} = 15 mA		0.45	V
	V_{OH}		I_{OH} = –1 mA	3.65		V
	V_{IL}				0.85	V
	V_{IH}			2.0		V
	I_{IL}				–0.35	mA
	I_{IH}				110	μA
	C_L				15	pF

Table 9-7b

EXTERNAL INTERFACE I/O MODULE DC CHARACTERISTICS

SIGNAL	SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	PARAMETERS		
				MIN.	MAX.	UNITS
STBØ/ → STB3/,	V_{OL}	Output Low Voltage	I_{OL} = 48 mA		0.4	V
OxDATy/,	V_{OH}	Output High Voltage	I_{OH} = –1.2 mA	2.4		V
	C_L	Capacitive Load			15	pF
(x=0,1,2,3; y=0 → 7)						
OINTx/,IINTx/	V_{OL}		I_{OL} = 16 mA		0.4	V
(x=0,1,2,3),	V_{OH}		I_{OH} = –5.2 mA	2.4		V
SYS RST/	C_L				15	pF
IxDATy/, ISBx,	V_{IL}	Input Low Voltage			0.85	V
OSTBx	V_{IH}	Input High Voltage		2.0		V
(x=0,1,2,3; y=0 → 7)	I_{IL}	Input Current at V_{IL}	V_{IL} = 0.45 V [1]		–5.25	mA
	I_{IH}	Input Current at V_{IH}	V_{IH} = 5.25 V [2]		260	μA
	C_L				15	pF
EXINT0/→ EXINT7/	V_{IL}				0.8	V
	V_{IH}			2		V
	I_{IL}		V_{IL} = 0.4 V [1]		–6.6	mA
	I_{IH}		V_{IH} = 2.4 V		40	μA
	C_L				15	pF

[1] Includes 5 mA due to a 1 K terminating resistor (this changes if the user changes resistor pack values).

[2] Includes 250 μA due to 1 K terminating resistor (which changes if resistor pack is changed).

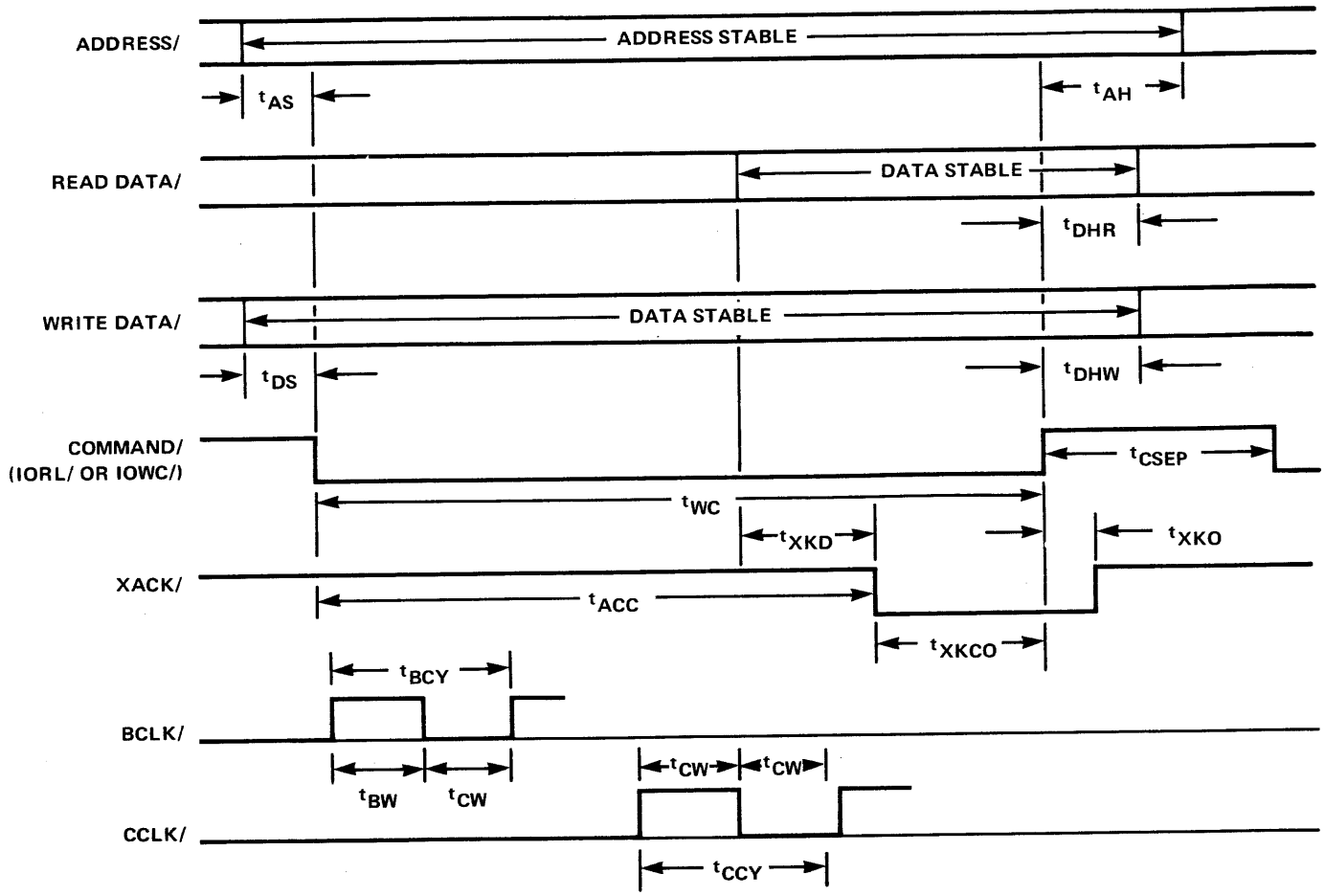


Figure 9-8. Command Timing

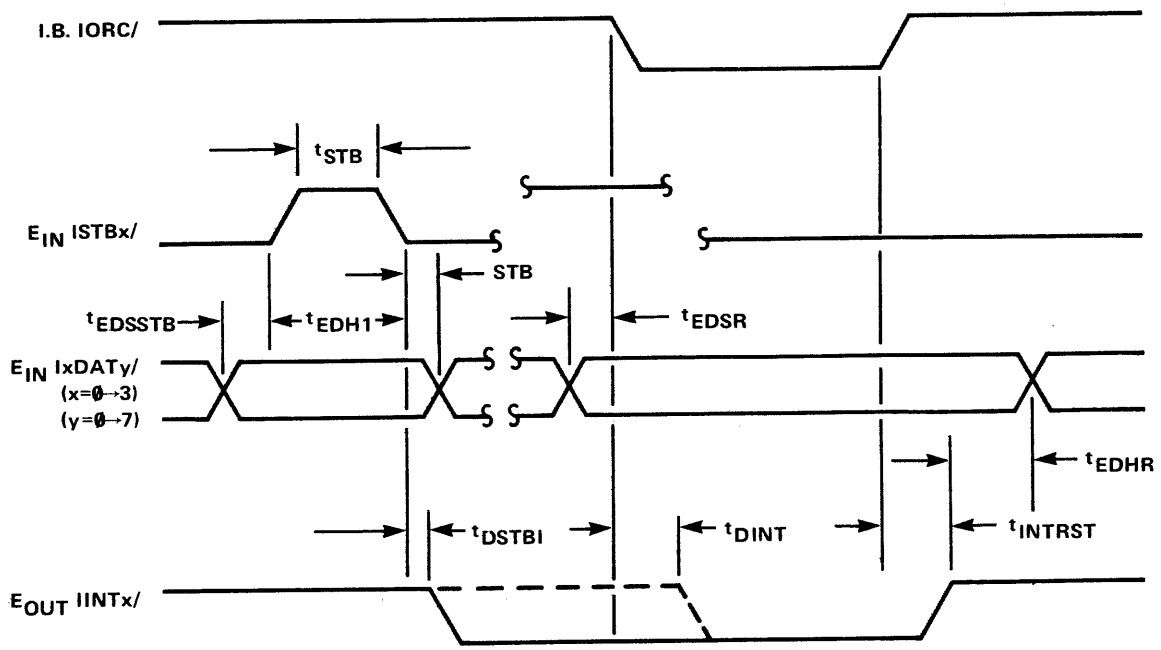


Figure 9-9a. External I/O Read Timing

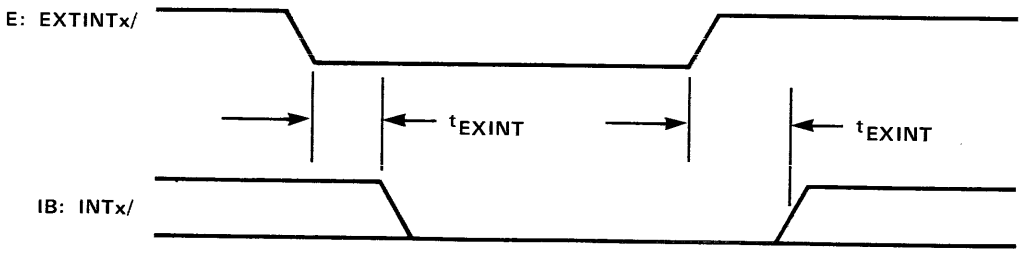


Figure 9-9b. Interrupt Timing

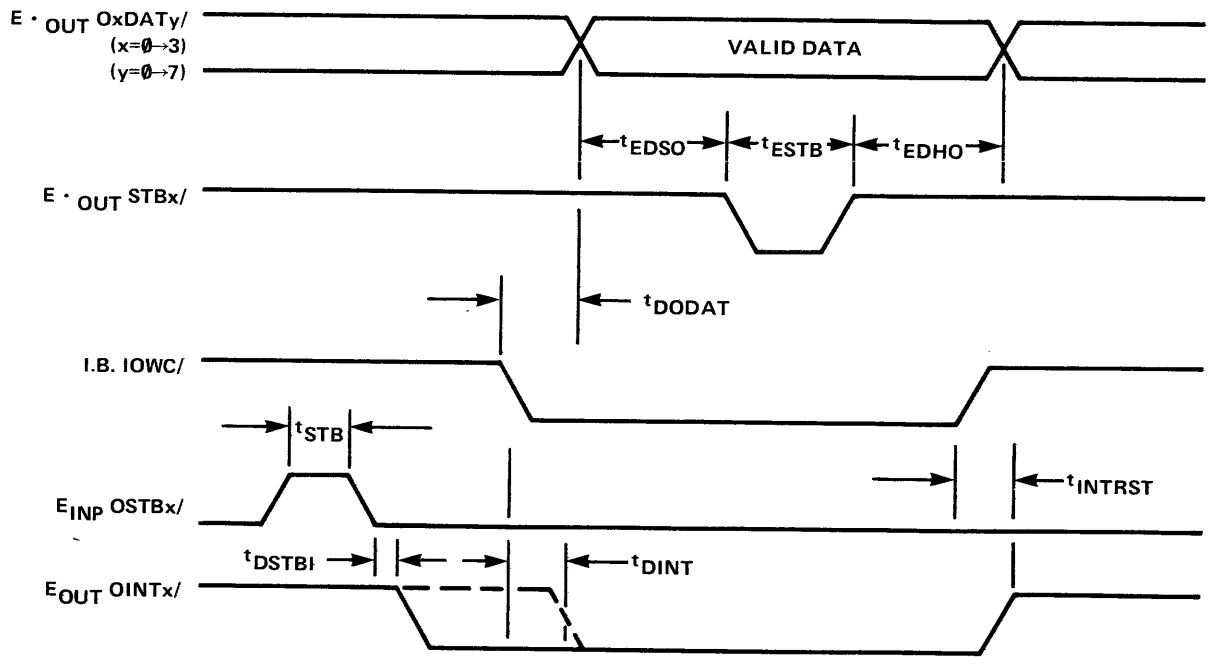


Figure 9-10. External I/O Write Timing

Chapter 10
INTELLEC MDS SYSTEM BUS

A significant measure of the INTELLEC MDS System's power and flexibility can be attributed to the design of its bus. The bus structure allows for multiple master-slave relationships between the various system modules. In fact, the bus can support eight masters in a parallel, priority network. By connecting adjacent master modules with a serial bus priority line, the maximum number of masters can be expanded to 16. In such a configuration, each master-pair contends for bus control via the parallel priority network. Once a pair is accorded bus control, the two masters in the pair further resolve contention via the serial priority line. This configuration allows for an increased number of master modules without incurring the timing overhead of a pure serial network. While a pure serial bus control network can be implemented on the INTELLEC MDS Bus, the maximum bus transfer rate of 5 MHz cannot be guaranteed in that application.

The bus provides its own clock which is derived independently from the processor clock. The bus clock provides a timing reference for resolving bus contention among multiple bus requests. This feature allows different speed processors to share resources on the same bus. Actual transfers via the bus, however, proceed asynchronously with respect to the bus clock. Thus, the transfer speed is dependent on the transmitting and receiving devices only. This design prevents slow master modules from being handicapped in their attempts to gain control of the bus, but does not restrict the speed at which faster modules can transfer data via the same bus. Once a bus request is granted, single or multiple read/write transfers can proceed at a maximum rate of 5 MHz (i.e., 5 million data words per second). The most obvious applications for the master-slave capabilities of the bus are multi-processor configurations and high-speed direct-memory-access (DMA) operations, but are by no means limited to these two.

The INTELLEC MDS System Bus (excluding power inputs) consists of 56 signal lines, including 16 address lines, 16 bidirectional data lines, and 8 multi-level interrupt lines. Thus, the system is capable of supporting 64K (65,536) words of storage. The data words can be 8 or 16 bits wide, allowing the system to support 8 or 16-bit processors. The address and data lines are driven by three-state devices, while the interrupt lines are open-collector driven.

10.1 BUS SIGNAL DESCRIPTIONS

This section defines each of the signal lines that comprise the INTELLEC MDS System Bus:

BCLK/ — *Bus clock*; used to synchronize bus control circuits on all master modules. BCLK/ has a period of 101.725 ns (9.8304 MHz frequency), 30%–70% duty cycle. BCLK/ may be slowed, stopped or single stepped, if desired.

INIT/ — *Initialization signal*; resets the entire system to a known internal state.

BPRN/ — *Bus priority in signal*; indicates to a particular master module, that no higher priority module is requesting use of the system bus. BPRN/ is synchronized with BCLK/.

BPRO/ — *Bus priority out signal*; used with serial (daisy chain) bus priority resolution schemes. BPRO/ is passed to the BPRN/ input of the master module with next lower bus priority.

BUSY/ — *Bus busy signal*; indicates that the bus is currently in use. BUSY/ prevents all other master modules from gaining control of the bus. BUSY/ is synchronized with BCLK/.

BREQ/ — *Bus request signal*; used with a parallel bus priority network to indicate

that a particular master module requires use of the bus for one or more data transfers. BREQ/ is synchronized with BCLK/.

MRDC/ — *Memory read command*; indicates that the address of a memory location has been placed on the system address lines and specifies that the contents of the addressed location are to be read and placed on the system data bus. MRDC/ is asynchronous with BCLK/.

MWTC/ — *Memory write command*; indicates that the address of a memory location has been placed on the system address lines and that a data word (8 or 16 bits) has been placed on the system data bus (note exceptions on CPU Module). MWTC/ specifies that the data word is to be written into the addressed memory location. MWTC/ is asynchronous with BCLK/.

IORC/ — *I/O read command*; indicates that the address of an input port has been placed on the system address bus and that the data at that input port is to be read and placed on the system data bus. IORC/ is asynchronous with BCLK/.

IOWC/ — *I/O write command*; indicates that the address of an output port has been placed on the system address bus and that the contents of the system data bus (8 or 16 bits) are to be output to the addressed port. IOWC/ is asynchronous with BCLK/.

XACK/ — *Transfer acknowledge signal*; the required response of a memory location or I/O port which indicates that the specified read/write operation has been completed. That is, data has been placed on, or accepted from, the system data bus lines. XACK/ is asynchronous with BCLK/.

INH1/ — *Inhibit RAM signal*; prevents any RAM memory device from responding to the memory address on the system address bus. INH1/ effectively allows ROM memory devices to override RAM devices when ROM and RAM memory occupy the same memory space (i.e., are assigned the same memory addresses).

INH2/ — *Inhibit ROM signal*; prevents any ROM memory device from responding to the memory address on the system address bus. INH2/ effectively allows auxiliary ROM (e.g., the bootstrap program) to override ROM devices when ROM and auxiliary ROM memory occupy the same memory space.

AACK/ — *Special acknowledge signal*; used with 8080 CPU-based systems. AACK/ is an advance acknowledge, in response to a memory read command, that allows the memory to complete the access without requiring the CPU to wait.

CCLK/ — *Constant clock:* provides a clock signal of constant frequency (9.8304 MHz) for use by the modules in the INTELLEC MDS System. CCLK/ coincides with BCLK/ and has a period of 101.725 ns, 30%–70% duty cycle.

INTR/ — *Direct interrupt signal*; dedicated spare provided to support coded interrupt requests in special applications of the system interrupt structure.

INTØ/–INT7/ — *8 Multi-level, parallel interrupt request lines*; used with a parallel interrupt resolution network. INTØ/ has highest priority, while INT7/ has lowest priority.

ADRØ/–ADRF/ — *16 Address lines*; used to transmit the address of the memory location or I/O port to be accessed. ADRF/ is the most significant bit.

DATØ/–DATF/ — *16 Bidirectional data lines*; used to transmit/receive information to/from

a memory location or I/O port. DATF/ is the most significant bit. In 8-bit systems, only lines DATØ/–DAT7/ are used (DAT7/ is the most significant bit).

10.2 PIN LIST

Table 10-1 provides pin allocations for each line of the system bus, as they are etched on the INTELLEC MDS System motherboard.

10.3 OPERATING CHARACTERISTICS

This section provides detailed information concerning the AC and DC requirements of the INTELLEC MDS Bus.

10.3.1 AC REQUIREMENTS

Table 10-2 and Figures 10-1 to 10-4 describe the INTELLEC MDS Bus timing requirements. These provide for data transfer rates up to 5 MHz.

10.3.2 DC REQUIREMENTS

Table 10-3 describes the INTELLEC MDS Bus recommended DC requirements.

Table 10-1

PIN ASSIGNMENT OF BUS SIGNALS ON INTELLEC® MOTHER BOARD

	(COMPONENT SIDE)			(CIRCUIT SIDE)		
	PIN	MNEMONIC	DESCRIPTION	PIN	MNEMONIC	DESCRIPTION
POWER SUPPLIES	1	GND	Signal GND	2	GND	Signal GND
	3	VCC	+ 5 VDC	4	VCC	+ 5 VDC
	5	VCC	+ 5 VDC	6	VCC	+ 5 VDC
	7	VDD	+ 12 VDC	8	VDD	+ 12 VDC
	9	VXI	Supply Spare 1	10	VXI	Supply Spare 1
	11	GND	Signal GND	12	GND	Signal GND
BUS CONTROLS	13	BCLK/	Bus Clock	14	INIT/	Initialize
	15	BPRN/	Bus Pri. In	16	BPRO/	Bus Pri. Out
	17	BUSY/	Bus Busy	18	BREQ/	Bus Request
	19	MRDC/	Mem Read Cmd	20	MWTC/	Mem Write Cmd
	21	IORC/	I/O Read Cmd	22	IOWC/	I/O Write Cmd
	23	XACK/	XFER Acknow	24	INH1/	Inhibit 1 disable RAM
				26	INH2/	Inhibit 2 disable PROM or ROM
SPARES	25	AACK/	Special	26		
	27			28		
	29			30		
	31	CCLK/	Constant Clock	32		
	33	INTR/	Direct Int	34		
INTERRUPTS	35	INT6/	Parallel	36	INT7/	Parallel
	37	INT4/	Interrupt	38	INT5/	Interrupt
	39	INT2/	Requests	40	INT3/	Requests
	41	INT0/		42	INT1/	
ADDRESS	43	ADRD/		44	ADRF/	
	45	ADRC/		46	ADRD/	
	47	ADRA/	Address	48	ADRB/	Address
	49	ADR8/	Bus	50	ADR9/	Bus
	51	ADR6/		52	ADR7/	
	53	ADR4/		54	ADR5/	
	55	ADR2/		56	ADR3/	
	57	ADR0/		58	ADR1/	
DATA	59	DATE/		60	DATF/	
	61	DATC/		62	DATD/	
	63	DATA/	Data	64	DATB/	Data
	65	DAT8/	Bus	66	DAT9/	Bus
	67	DAT6/		68	DAT7/	
	69	DAT4/		70	DAT5/	
	71	DAT2/		72	DAT3/	
	73	DAT0/		74	DAT1/	
POWER SUPPLIES	75	GND	Signal GND	76	GND	Signal GND
	77	VBB	-10 VDC	78	VBB	-10 VDC
	79	VX2	-12 VDC	80	VX2	-12 VDC
	81	VCC	+5 VDC	82	VCC	+5 VDC
	83	VCC	+5 VDC	84	VCC	+5 VDC
	85	GND	Signal GND	86	GND	Signal GND

Table 10-2

INTELLEC® MDS BUS AC REQUIREMENTS

PARAMETER	MIN.	MAX.	DESCRIPTION	REMARKS
t_{BCY}	100ns		Bus clock period	
t_{BW}	0.3 xt_{BCY}	0.7xt_{BCY}	Bus clock width	
t_{AS}	50ns		Address set up time	Relative to active command
t_{DS}	50ns		Write data set up time	Relative to active command
t_{AH}	50ns		Address hold time	Relative to command removal
t_{DH}	50ns		Write data hold time	Relative to command removal
t_{DXL}	0ns		Read data set up time	Relative to acknowledge (XACK/)
t_{DXT}	0ns		Read data hold time	Relative to command removal
t_{CX}	0ns		Acknowledge hold time	Relative to command removal
t_{ACC}	0ns	t_{ACCBT}	Acknowledge delay	1 The max. is imposed only if the bus timeout feature is engaged (a field option)
t_{ACCBT}	5ns	15ns		
t_{CMD}	100ns		Command pulse width	
t_{CI}		100ns	Inhibit delay	Relative to address
t_{ACCB}	1.5 μsec 2		Acknowledge of inhibiting slave	2 The inhibited slave must be allowed to terminate any irreversible timing operations initiated by detection of a valid command prior to its inhibit. This is accomplished by insuring that the inhibiting slave acknowledges the command after the latest possible acknowledgement (t_{ACCA}) from the inhibited slave. For the MDS dynamic memory, this requires the stated minimum.

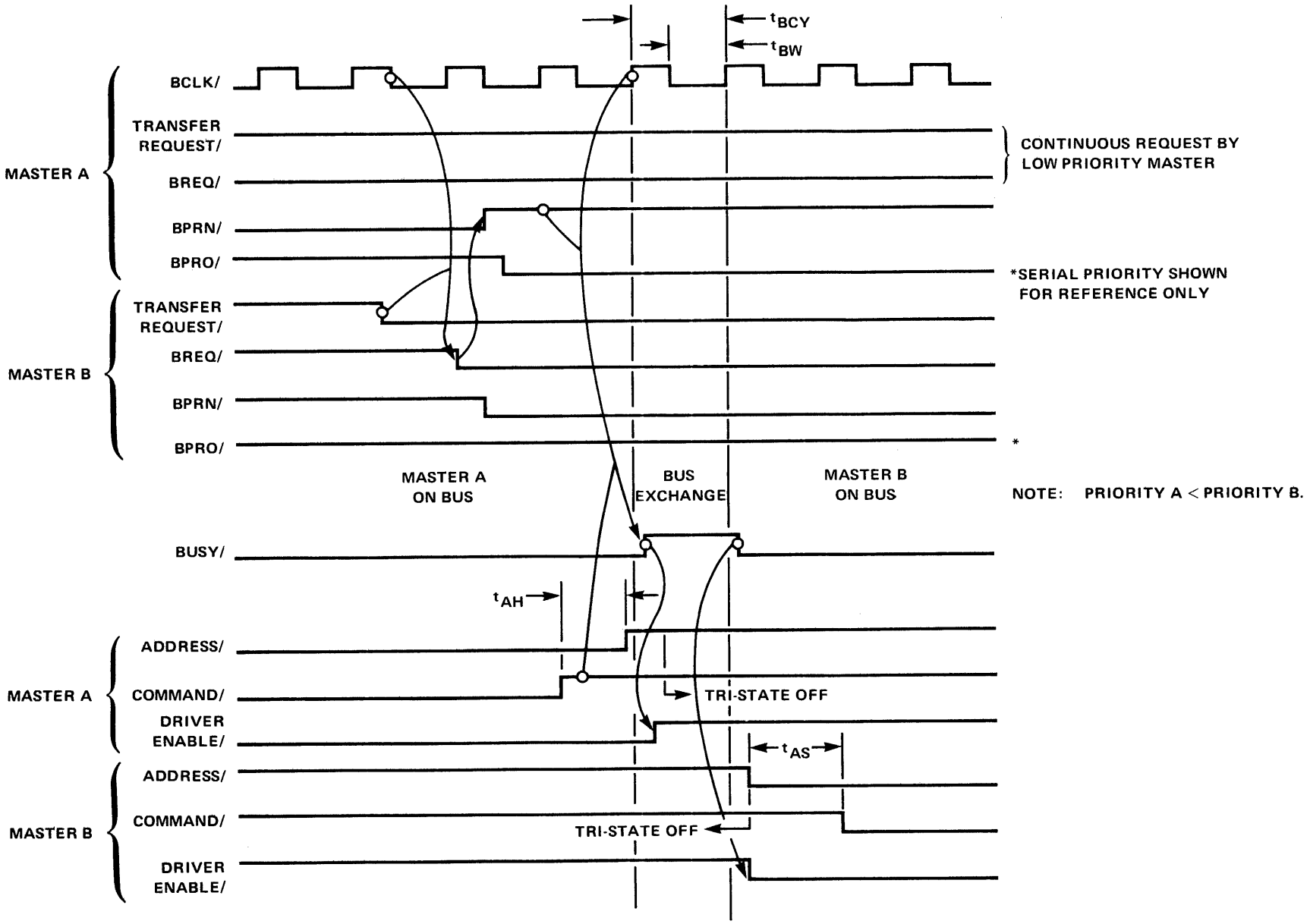


Figure 10-1. Bus Exchange Between Masters

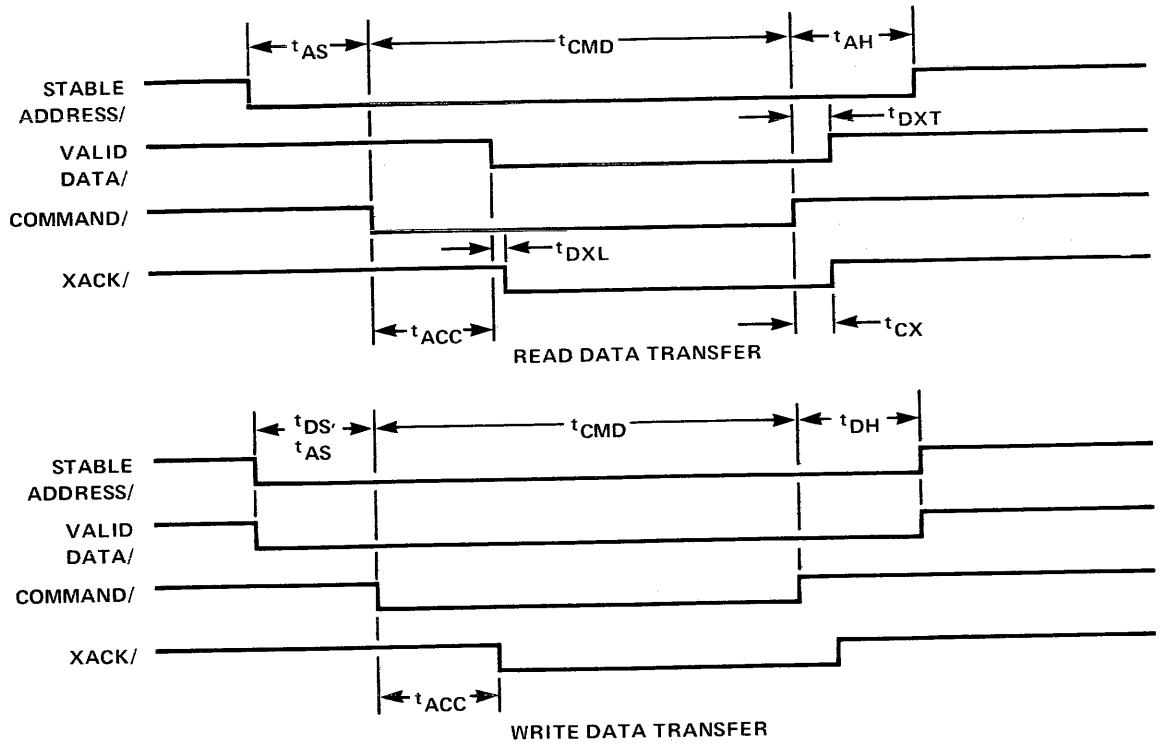


Figure 10-2. Data Transfer

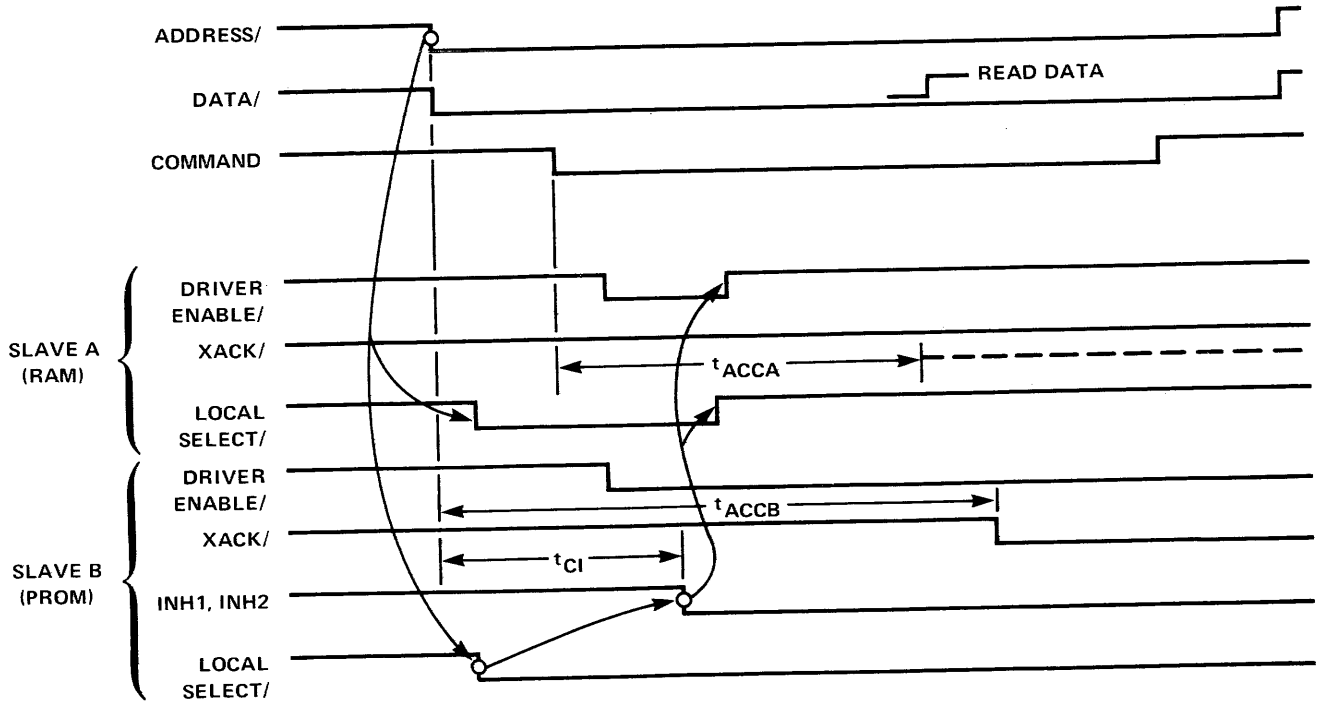


Figure 10-3. Inhibit Operation

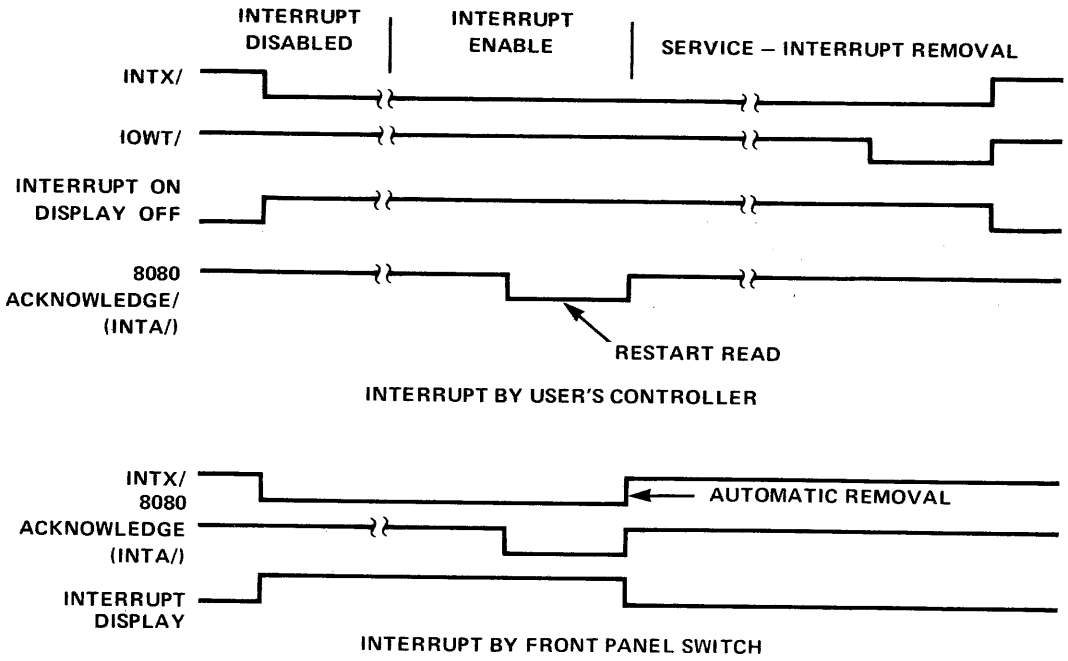


Figure 10-4. Interrupt by Front Panel Switch

Table 10-3

INTELLEC® MDS BUS DC REQUIREMENTS

BUS SIGNALS	DRIVER		LOAD PER BOARD		PULL-UP/DOWN RESISTOR
	LOCATION	DRIVE (MIN.)	LOCATION	SOURCING (MAX.)	
INIT/	FP	TTL, 32 mA	All	1.8 mA	None
BCLK/, CCLK/	FP	TTL, 48 mA	Master	HTTL, 2 mA	220/330Ω termination on Mother Board
BREQ/	Master	TTL, 16 mA	FP	1.6 mA	1 KΩ pull-up at FP
BPRN/	FP/Master	TTL, 16 mA	Master	1.6 mA	None
BPRO/	Master	TTL, 3.2 mA	Master	1.6 mA	None
BUSY/	Master	OR, 20 mA	Master	2 mA	1.0 KΩ pull-up at FP
MRDC/, MWTC/	Master	TRI, 32 mA	16KB-RAM 6KB-PROM	4 mA 1.6 mA	1.1 KΩ pull-up at FP
IORC/, IOWC/	Master	TRI, 32 mA	I/O board	1.6 mA	1.1 KΩ pull-up at FP
XACK/, AACK/	Slave	TRI, 16 mA	Master	1.6 mA	510 KΩ pull-up at FP
DATF/–DATØ/	Master	TRI, 15 mA	Slave	0.5 mA	2.2 KΩ pull-up at FP
ADRF/–ADRØ/	Master	TRI, 15 mA	Slave	0.5 mA	2.2 KΩ pull-up at FP
INH1/, INH2/	All	OR, 16 mA	RAM	1.6 mA	1 KΩ pull-up at FP
INT7/–INTØ/	All	OR, 16 mA	Master	1.6 mA	1 KΩ pull-up at FP

NOTES:

1. Input voltage levels: High 2.4 to 5.0 volts
   Low 0.0 to 0.8 volts
2. Output voltage level: High 2.0 to 5.25 volts
   Low 0.0 to 0.45 volts
   OR – open collector
   TTL – totem pole output
   TRI – three-state
3. Leakage current of an input ⩽ 40 a
   Leakage current of an output ⩽ 100 a
4. FP: Front Panel Control Board
   Master: CPU board, DMA boards and Front Panel Control Board
   Slave: Memory boards, I/O boards, DMA boards and Front Panel Control Board
5. Maximum number of Master devices = 9
   Maximum number of PC boards in an INTELLEC MDS System = 18

Chapter 11
CHASSIS, MOTHERBOARD AND POWER SUPPLIES

The INTELLEC MDS chassis, motherboard and power supplies are designed to provide the housing, interconnection and power services which bring separate circuit cards together as a fully operational INTELLEC MDS System. Since these three components of the INTELLEC MDS System are, essentially, quite simple, they will not be described in much detail.

The INTELLEC MDS System includes two OEM power supplies. One transforms 115 VAC or 230 VAC (field-selectable), 50–60 Hz power obtainable from a wall socket, into +5 VDC regulated power for use by the system. The other supply transforms standard 115/230 VAC, 50–60 Hz power into +12 VDC, –10 VDC and –12 VDC regulated power levels for use by the system. Table 11-1 summarizes the specifications for both power supplies. Their locations within the INTELLEC MDS System are illustrated in Figure 11-1.

The motherboard is, simply, a printed circuit board which has mounted on it the connectors which hold the various cards in the system. The motherboard is located in the bottom plane of the chassis. The chassis provides 18 card slots. Card position 1 must be occupied by the Front Panel Control Module, while card position 2 must be occupied by the primary Central Processor Module. The other modules in the system may reside in any card position, with one stipulation: bus master modules must be in odd-numbered card positions. A bus master module may only be placed in an even-numbered slot if it is paired to another master module in an adjoining odd position [by paired we mean that the BPRO/ output (bus priority out) from the odd master must be connected to the BPRN/ input (bus priority in) of the even master].

The motherboard is connected to the 86-pin, P1 connectors on each module in the system. In addition, the motherboard is connected to the 60-pin P2 auxiliary connectors on the Front Panel Control Module (card slot 1) and the primary Central Processor Module (card slot 2).

Auxiliary connectors are intended for use on an incremental basis, as required. Available options include a single wire-wrap connector and a dual PCB connector with one-to-one etched wiring between connectors. Mounting hardware is provided with options for easy assembly.

Table 11-1

POWER SUPPLIES: SPECIFICATIONS

PARAMETERS	+5 VDC	+12 VDC	–10 VDC	–12 VDC	CONDITIONS
Maximum Load	35A	3A	3A	0.25A	–
Line Regulation	±0.01%	±0.01%	±0.01%	±0.01%	±10% input variation
Load Regulation	±0.02%	±0.02%	±0.02%	±0.02%	from no load to max load
Maximum Ripple	10 MV PP	10 MV PP	10 MV PP	10 MV PP	–
Transient Response	50 μsec max.	30 μsec max.	30 μsec max.	30 μsec max.	50% load change
Overvoltage Protection	6.0 to 6.3 V	±15 V ±1 V	±13 V ±1 V	–	remote sensing
Temperature Range	*Full ratings apply in minimum 75 CFM moving air at 0–65°C; 50–60 Hz.				
Humidity	90% maximum relative with no condensation				

*The INTELLEC® MDS system includes fans which provide the necessary ventilation.

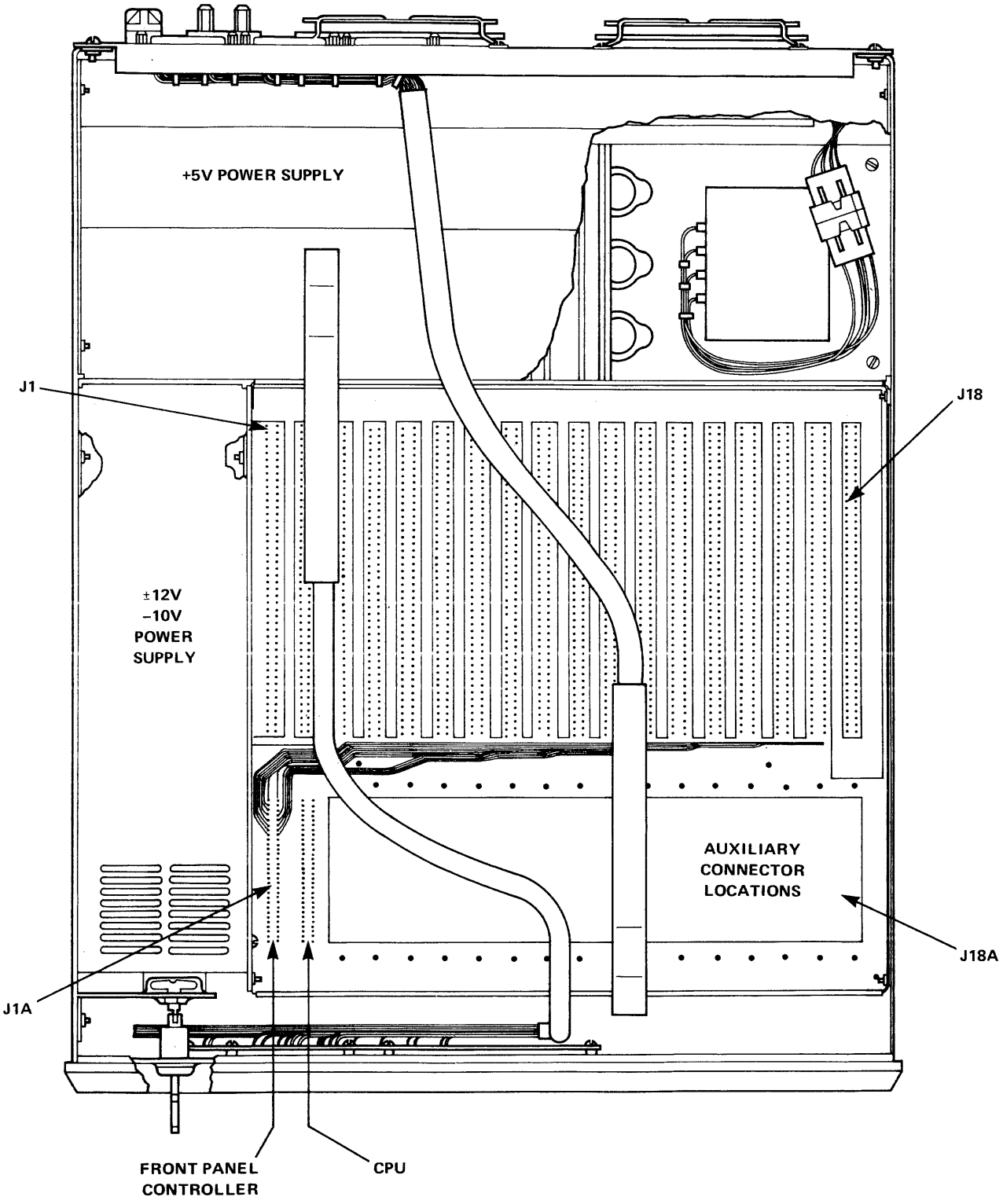
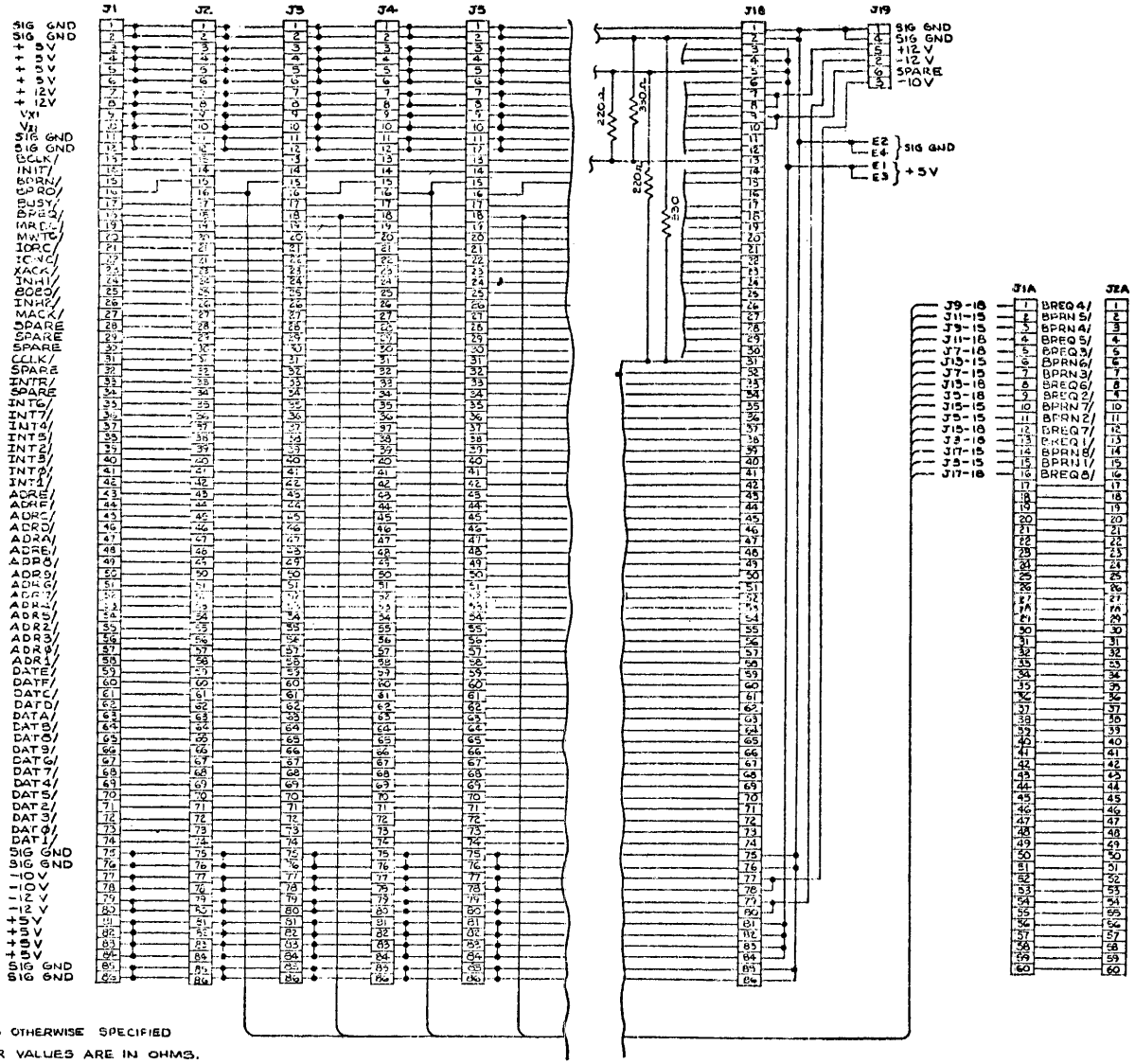
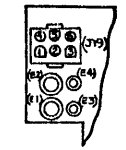


Figure 11-1. INTELLEC® MDS Chassis





PIN ORIENTATION
CONNECTOR SIDE
J19, E1, E2, E3, E4.

NOTE: UNLESS OTHERWISE SPECIFIED
1. RESISTOR VALUES ARE IN OHMS.
2. LATEST ARTWORK REVISION IS 3.

intel © 3065 BOWERS AVE. SANTA CLARA CALIF. 95051

TITLE			
SCHEMATIC MOTHERBOARD			
SIZE	DEPT	DRAWING NO.	REV
D	410	E2000307	2.

1

Figure 11-2. Motherboard Schematic

Chapter 12
INTELLEC MDS SYSTEM UTILIZATION

This chapter provides the information necessary to install and use the INTELLEC MDS System. Because the INTELLEC MDS is delivered in a ready-to-use condition, no special procedures are required to install the basic system. The user must, however, select the proper AC input level (115V or 230V) on the rear panel of the INTELLEC MDS cabinet, prior to applying AC power. Note that fuse ratings differ for the two AC power levels; appropriate fusing should be provided.

The only site-preparation consideration is that of the environment and those specifications are standard. The INTELLEC MDS System can be operated in any location where the temperature remains within the range of 0–55°C, and relative humidity does not exceed 90%, non-condensing.

The one area requiring the user's attention is that of connecting the basic system to the desired I/O peripherals. Section 12.1 provides the necessary information on system I/O interfacing. Cabling instructions, dedicated I/O port addresses, external driver/receiver requirements and INTELLEC MDS back panel pin information are all provided in this section. Section 12.2 lists the simple procedures for modifying the ASR-33 Teletype for use with the INTELLEC MDS System.

The standard INTELLEC MDS Monitor Module provides six dedicated I/O interfaces. In addition, Intel offers a Direct Memory Access (DMA) Module, with five input ports and five output ports for high-speed transfers, an Input/Output (I/O) Module, with four input ports and four output ports, a Diskette Controller, a ROM Simulator and Intelligent Chip Emulator (ICE) Modules, as options to the basic INTELLEC MDS System. These modules provide a variety of expanded capabilities. The system's I/O provisions do not, however, have to be limited to those provided by Intel modules only. A user who wishes to design his own I/O controllers for use within the INTELLEC MDS System should refer to Chapter 10 for system bus interfacing requirements.

12.1 SYSTEM I/O INTERFACING

The INTELLEC MDS Monitor Module includes six dedicated I/O interfaces for the following devices:

- Teletype, including its paper tape reader
- Cathode ray tube (CRT) terminal or other compatible device (TTL or RS232 interface is jumper-selectable)
- High-speed paper tape reader
- High-speed paper tape punch
- Line printer
- Universal PROM Programmer

Table 12-1 lists those I/O port addresses which are dedicated to the interfaces on the Monitor Module, as well as those port addresses which are dedicated to use by the Front Panel Control Module. The remaining I/O addresses (256 maximum) are available for use at the designer's discretion.

The optional DMA and I/O Modules further enhance the system's I/O capabilities. The ports on these two modules do not, however, have dedicated port addresses. Instead, each module is assigned a BASE address by the user. The BASE address then defines the unique 8-bit address for each port (refer to Chapters 8 and 9 for BASE address selection procedures).

Table 12-2 lists the driver/receiver requirements for each I/O port (interface) provided by the Monitor, DMA and I/O Modules.

Cabling

The Monitor, DMA and I/O Modules all communicate with their associated devices via the 100-pin J1 connector on top of each printed circuit board. Because the Monitor Module is a standard component, the INTELLEC MDS System is delivered with the Monitor Module already connected to six connectors mounted on the back panel of the INTELLEC chassis. A harness connects the

Table 12-1

SUMMARY OF DEDICATED I/O ADDRESSES FOR INTELLEC® MDS SYSTEM

ADDRESS	INPUT	OUTPUT
00FF	*Real Time Clk	*Enable RTC
00FE	Reserved	**Override
00FD	Reserved	**Store Cur Level
00FC	**Int Mask	**Int Mask
00FB	LPT Status	LPT Control
00FA	INT Status	LPT Data
00F9	PT Status	PT Control
00F8	PTR Data	PTP Data
00F7	CRT Status	CRT/Control
00F6	CRT Data	CRT Data
00F5	TTY Status	TTY Control
00F4	TTY Data	TTY Data
00F3	–	Monitor Int. Cntrl
00F2	–	PROM prog address LSB
00F1	PROM prog status	PROM prog high addr/control
00F0	PROM prog data	PROM prog DATA

**Implemented on Front Panel Control Module in INTELLEC MDS system.

**Implemented on CPU Module.

module's J1 connector with six bulkhead-mounted connectors, labeled PROM, CRT, PTR, LPT, TTY, and PTP. Consequently, to install any of the I/O devices supported by the Monitor Module, the user merely plugs the connector-cable assembly from the device into the appropriate connector on the INTELLEC MDS back panel. Figure 12-2 correlates each pin on the Monitor Module's J1 connector with the pins on the connectors at both ends of the Monitor harness and, in turn, with the pins on the connectors mounted on the back panel of the INTELLEC MDS chassis. Figure 12-2 also lists the cable assemblies to be used with each of the six I/O devices.

To connect an external I/O device to the DMA Module or the I/O Module, the user must run the cable from the external device through the foam-protected slot at the top of the INTELLEC back panel and attach it through an appropriate connector to the 100-pin J1 edge connector on the top of the DMA or I/O Module. A Viking 3V450/1JN5 and protective hood assembly (SAE 8030 type) is a suitable cable-connector assembly for joining a device to the J1 connector on the DMA or I/O Module. Table 12-3 lists pin allocations on the J1 connector of the DMA Module, while Table 12-4 provides the same information for the J1 connector of the I/O Module.

12.2 TELETYPE MODIFICATIONS

The ASR-33 Teletype must receive the following internal modifications and external connections, for use with the INTELLEC MDS System.

Internal Modifications

(1) The current source resistor value must be changed to 1450Ω. This is accomplished by moving a single wire (see Figure 12-7).

(2) A full duplex hook-up must be created internally. This is accomplished by moving two wires on a terminal strip (see Figures 12-6 and 12-9).

(3) The receiver current level must be changed from 60 mA to 20 mA. This is accomplished by moving a single wire (see Figures 12-6 and 12-9).

(4) A relay circuit must be introduced into the paper tape reader drive circuit. The circuit consists of a relay, resistor, a diode, a thyractor and a suitable mounting fixture. This change requires the assembly of a small "vector" board with the relay circuit on it. It may be mounted in the Teletype by using two tapped holes in the base plate (see Figure 12-3). The relay circuit may then be added without alteration of the existing circuit (see Figures 12-4, 12-5 and 12-6). That is, wire "A" (figure 12-9), to be connected to the brown wire in Figure 12-4 may be spliced into the brown wire near its connector plug. The "line" and "local" wires must then be connected to the mode switch (see Figures 12-7 and 12-9).

Table 12-2

I/O DEVICE DRIVER/RECEIVER REQUIREMENTS

MODULE	I/O PORT (INTERFACE)	EXTERNAL DEVICE DATA DRIVER/RECEIVER REQUIREMENTS
MONITOR MODULE	Teletype, ASR 33	Current loop, 20 mA
	CRT	8093 TTL drivers capable of sinking 32 mA, 8251 TTL receivers with 1000 ohm pull-up resistors OR RS232 interface (TTL and RS232 are jumper-selectable)
	Paper Tape Reader	7437 TTL drivers capable of sinking 48 mA 8097 TTL receivers without pull-up
	Paper Tape Punch	8097 TTL drivers capable of sinking 32 mA 8097 TTL receivers with 470 ohm pull-up resistors
	Line Printer	7437 TTL Command drivers capable of sinking 48 mA 8098 TTL data driver capable of sinking 32 mA
	Universal PROM Programmer	8097 TTL receivers without pull-up resistors, 7437 TTL command drivers capable of sinking 48 mA 8098 TTL data driver capable of sinking 32 mA
DMA MODULE	Output Ports	7437 TTL drivers capable of sinking 48 mA
	Input Ports	74S257 TTL receivers with 150 ohm pull-up resistors
I/O MODULE	Output Ports	7437 TTL drivers capable of sinking 48 mA
	Input Ports	8212 TTL receivers with optional pull-up resistors

External Connections

(1) A two-wire receive loop must be created. This is accomplished by the connection of two wires between the Teletype and the SYSTEM in accordance with Figure 12-9.

(2) A two-wire send loop similar to the receive loop must be created. (See Figure 12-9.)

(3) A two-wire tape reader loop connecting the reader control relay to the SYSTEM must be created. (See Figure 12-9.)

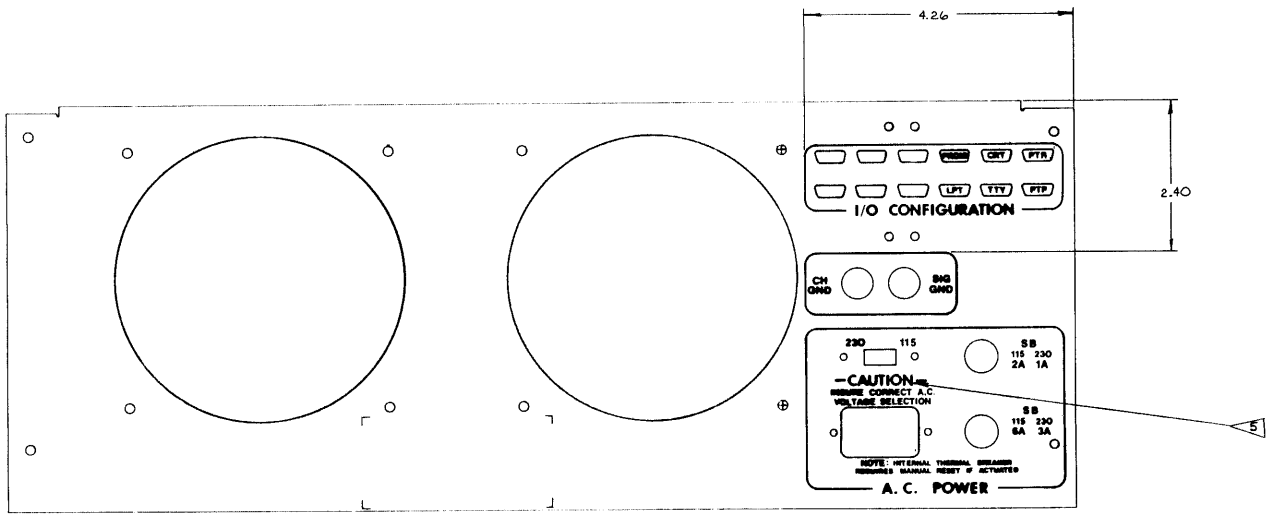


Figure 12-1. INTELLEC® MDS Back Panel

SCALE: 1/1	SIZE	DEPT	DRAWING NO.	REV
SHEET 2 OF 2	D	410	3000390	B

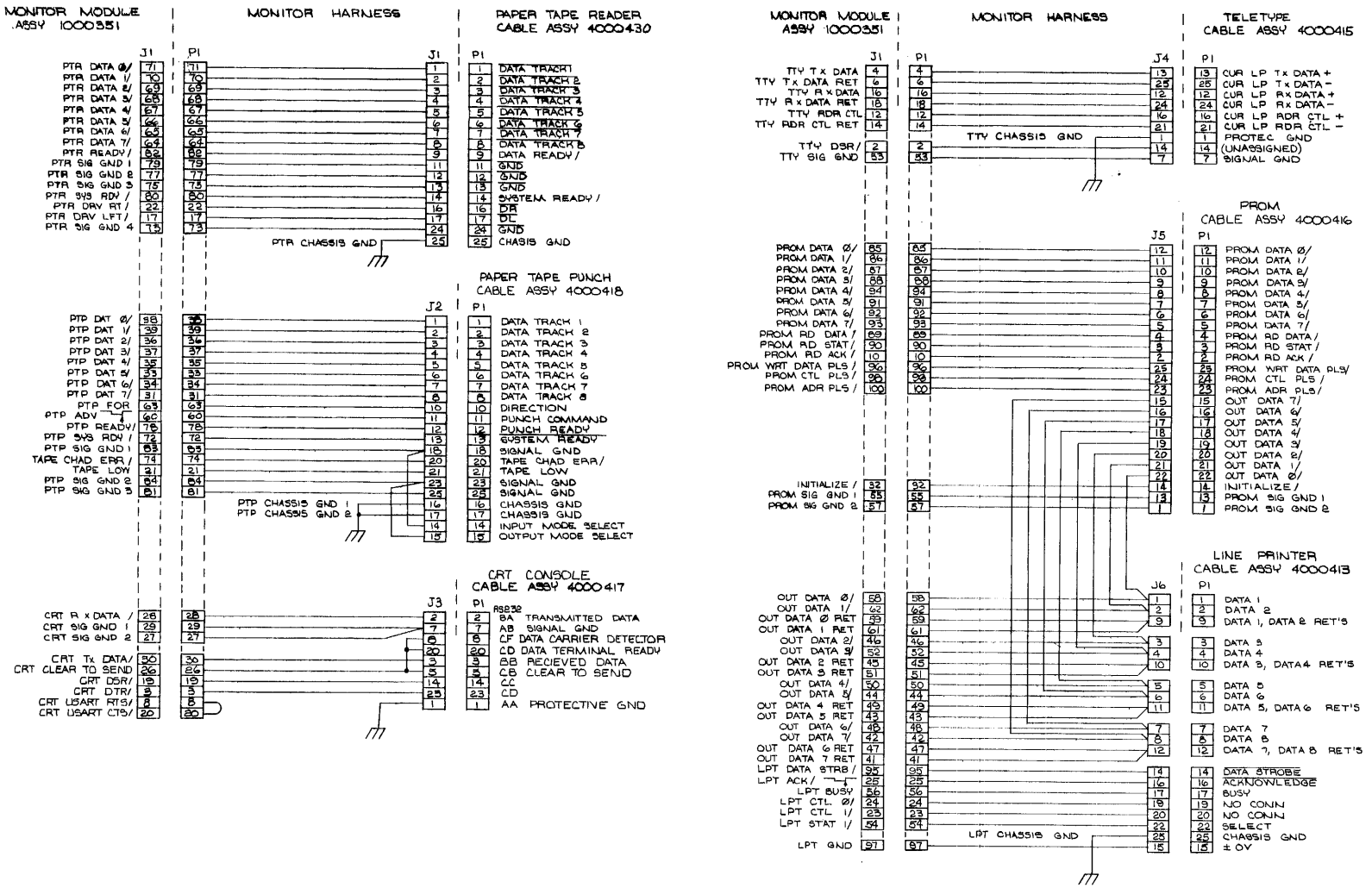


Figure 12-2. INTELLEC® MDS Back Panel Pin List

intel® © INTEL 1978	3065 BOWERS AVE. SANTA CLARA CALIF. 95051		
TITLE: SCHEMATIC MONITOR HARNESS			
SIZE: D	DEPT: 410	DRAWING NO.: 2000377	REV: 0

Table 12-3

DMA MODULE J1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	XFER RQ/	Transfer request
2	XFER DIR OUT	Transfer direction out
3	EX INTERRUPT/	External interrupt
4	XFER ACKNOWLEDGE/	Transfer acknowledge
5		
6	XFER DIR IN	Transfer direction in
7		
8	DELAY INT/	Delay interrupt
9		
10	OUTPORT TAG/	Output strobe
11		
12	INPORT TAG/	Input strobe
13		
14	RESET INTERRUPT/	Reset interrupt
15		
16	INPORTØ/	Input strobes
17		
18	INPORT1/	
19		
20	INPORT2/	
21		
22	INPORT3/	
23		
24	TAG3/	Tag register, bit 3
25		
26	SYS RESET/	System reset
27		
28		
29		
30		
31		
32		
33		
34		
35		
36	TAG2/	Tag register, bit 2
37		
38	TAG1/	Tag register, bit 1
39		
40	TAGØ/	Tag register, bit 0
41		
42	STATUS 3/	External status, bit 3
43		
44	STATUS 2/	External status, bit 2
45		
46	STATUS 1/	External status, bit 1
47		
48	STATUS Ø/	External status, bit 0
49		
50	OUTPORT 3/	

Table 12-3

DMA MODULE J1 CONNECTOR PIN LIST (continued)

PIN	SIGNAL	FUNCTION
51		
52	OUTPORT2/	Output strobes
53		
54	OUTPORT1/	
55		
56	OUTPORTØ/	
57		
58		
59		
60	DATA IN7/	Data input bus (from device)
61		
62	DATA IN6/	
63		
64	DATA IN5/	
65		
66	DATA IN4/	
67		
68	DATA IN3/	
69		
70	DATA IN2/	
71		
72	DATA IN1/	
73		
74	DATA INØ/	
75		
76	DATA OUT3/	Data output bus (to device)
77		
78	DATA OUT2/	
79		
80	DATA OUTØ/	
81		
82	DATA OUT1/	
83		
84	DATA OUT4/	
85		
86	DATA OUT7/	
87		
88	DATA OUT6/	
89		
90	DATA OUT5/	
91		
92	200 ns	Test
93		
94	400 ns	Test points
95		
96	800 ns	
97		
98	1600 ns	
99		
100	ASSERT RETAIN BUS/	

Table 12-4

I/O MODULE J1 CONNECTOR PIN LIST

PIN	SIGNAL	FUNCTION
1	GND	Ground
2	GND	
3	OØDAT6/	Data out bus from output portØ, bit6
4	GND	Ground
5	OØDAT3/	Data out bus from output port 0
6	OØDAT1/	
7	OØDATØ/	
8	OØDAT2/	
9	OØDAT7/	
10	OØDAT5/	
11	OØDAT4/	
12	IINT3/	Interrupt from input port 3
13	OINTØ/	Interrupt from output port 0
14	OIDAT6/	Data out bus from output port 1
15	OIDAT3/	
16	IINT1/	Interrupt from input port 1
17	OINT1/	Interrupt from output port 1
18	OIDAT1/	Data out bus from output port 1
19	OIDATØ/	
20	OIDAT2/	
21	OIDAT7/	
22	OIDAT5/	
23	OIDAT4/	
24	O2DAT6/	Data out bus from output port 2
25	O2DAT3/	
26	O2DAT1/	
27	O2DATØ/	
28	IINT2/	Interrupt from input port 2
29	IINTØ/	Interrupt from input port 0
30	OINT2/	Interrupt from output port 2
31	O2DAT2/	Data out bus from output port 2
32	O2DAT7/	
33	OINT3/	Interrupt from output port 3
34	O2DAT5/	Data out bus from output port 2
35	O2DAT4/	
36	O3DAT6/	Data out bus from output port 3
37	O3DAT3/	
38	O3DAT1/	
39	O3DATØ/	
40	OSTB3	Output strobes from devices
41	OSTBØ	
42	OSTB1	
43	OSTB2	
44	ISTB3	Input strobes from devices
45	ISTB2	
46	ISTB1	
47	ISTBØ	
48	EXTINT1/	External interrupt requests
49	EXTINTØ/	
50	O3DAT2/	Data out bus from output port 3

Table 12-4

I/O MODULE J1 CONNECTOR PIN LIST (continued)

PIN	SIGNAL	FUNCTION
51	O3DAT7/	Data out bus from output port 3
52	O3DAT4/	
53	O3 DAT5/	
54	SYS RST/	System reset
55	IØDAT1/	Data in bus to input port 0
56	IØDATØ/	
57	IØDAT3/	
58	IØDAT2/	
59	IØDAT4/	
60	IØDAT5/	
61	IØDAT7/	
62	IØDAT6/	
63	I1DAT2/	Data in bus to input port 1
64	I1DAT3/	
65	I1DATØ/	
66	I1DAT1/	
67	I1DAT5/	
68	I1DAT4/	
69	I1DAT7/	
70	I1DAT6/	
71	I2DAT2/	Data in bus to input port 2
72	I2DAT3/	
73	STBØ/	Output strobes to devices
74	STB1/	
75	I2DATØ/	Data in bus to input port 2
76	I2DAT1/	
77	I2DAT6/	
78	I2DAT7/	
79	I2DAT4/	
80	STB2/	Output strobes to devices
81	STB3/	
82	T2DAT5/	Data in bus to input port 2
83	EXTINT2/	External interrupt requests
84	EXTINT3/	
85	EXTINT5/	
86	EXTINT4/	
87	EXTINT6/	
88	EXTINT7/	
89	I3DAT1/	Data in bus to input port 3
90	I3DATØ/	
91	I3DAT3/	
92	I3DAT2/	
93	GND	Ground
94	GND	
95	I3DAT6/	Data in bus to input port 3
96	I3DAT7/	
97	I3DAT4/	
98	I3DAT5/	
99	GND	Ground
100	GND	

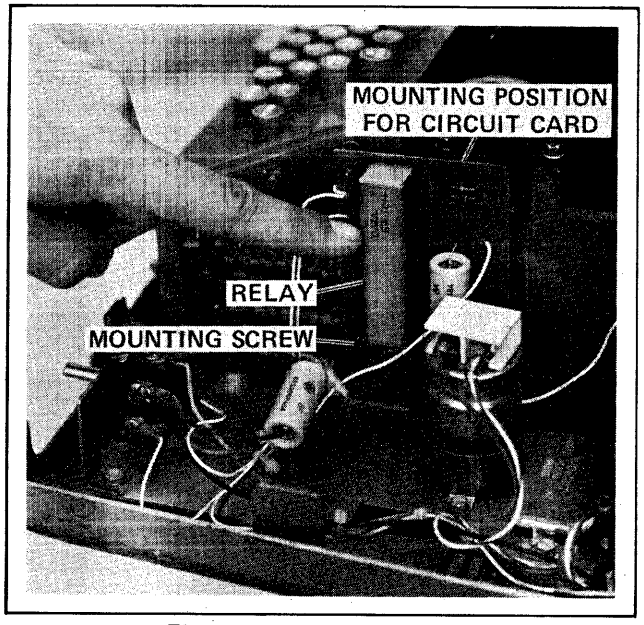


Figure 12-3. Relay Circuit (Alternate)

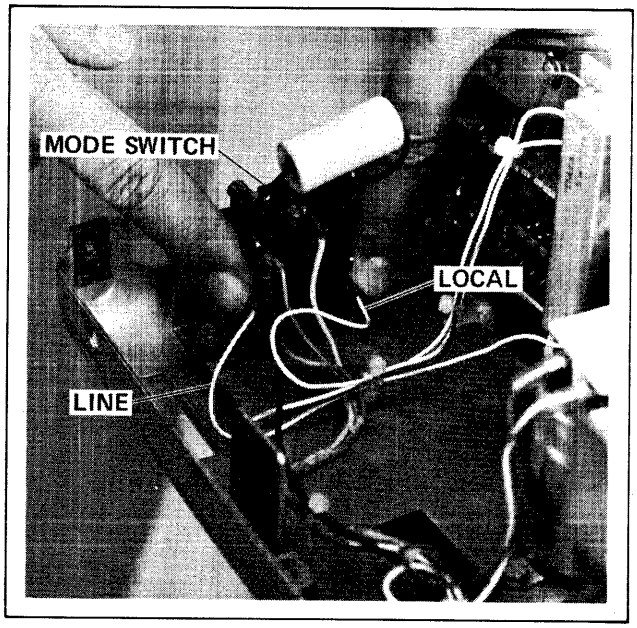


Figure 12-5. Mode Switch

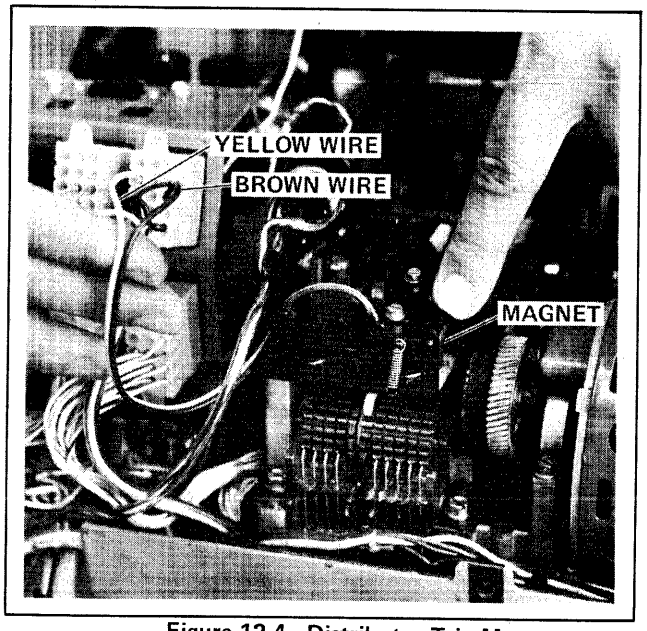


Figure 12-4. Distributor Trip Magnet

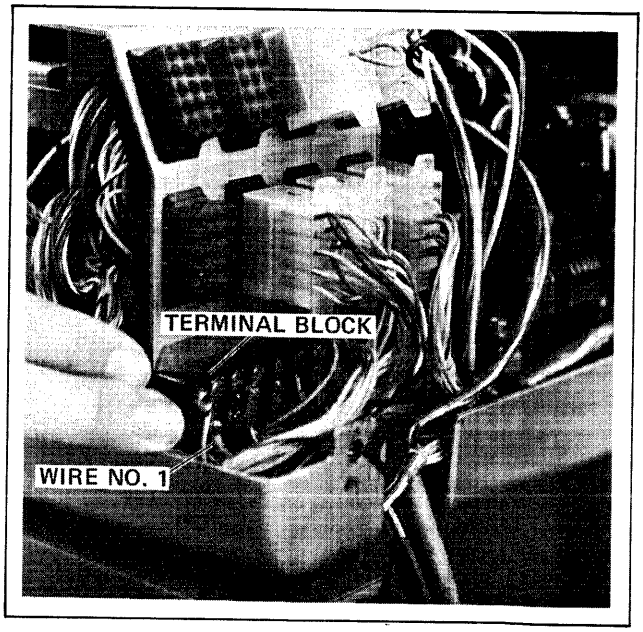


Figure 12-6. Terminal Block

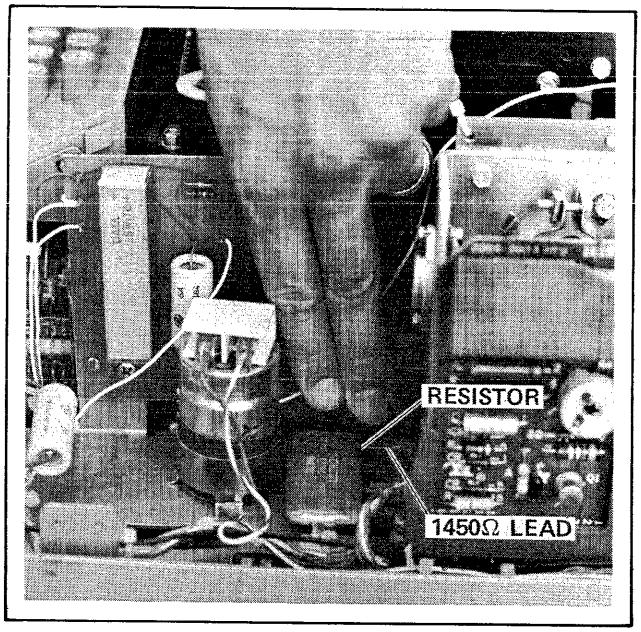


Figure 12-7. Current Source Resistor

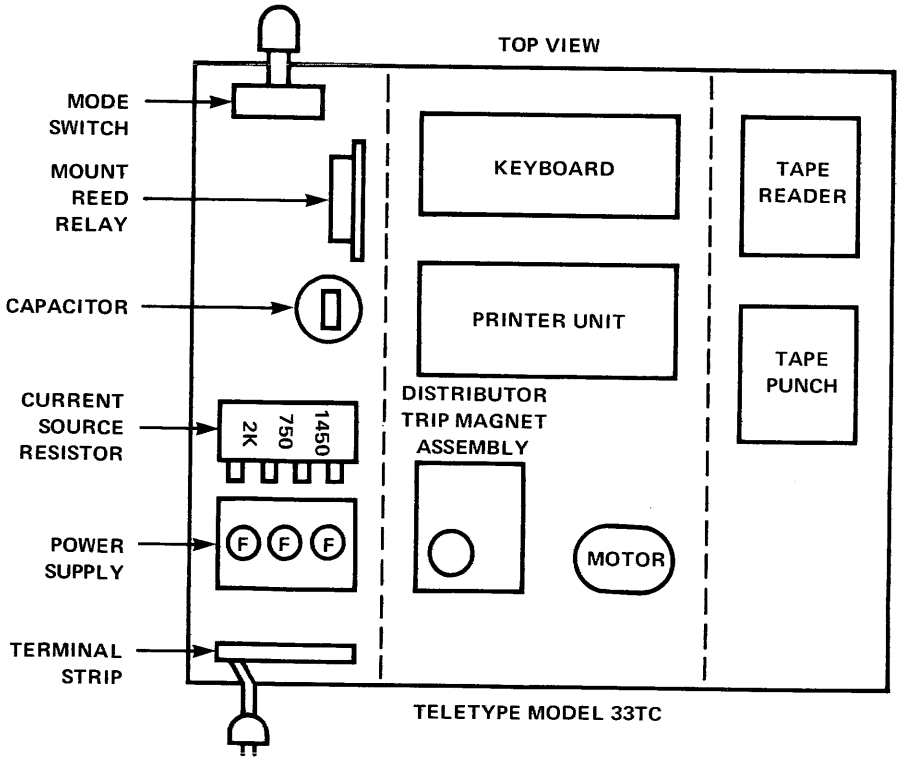


Figure 12-8. Teletype Layout

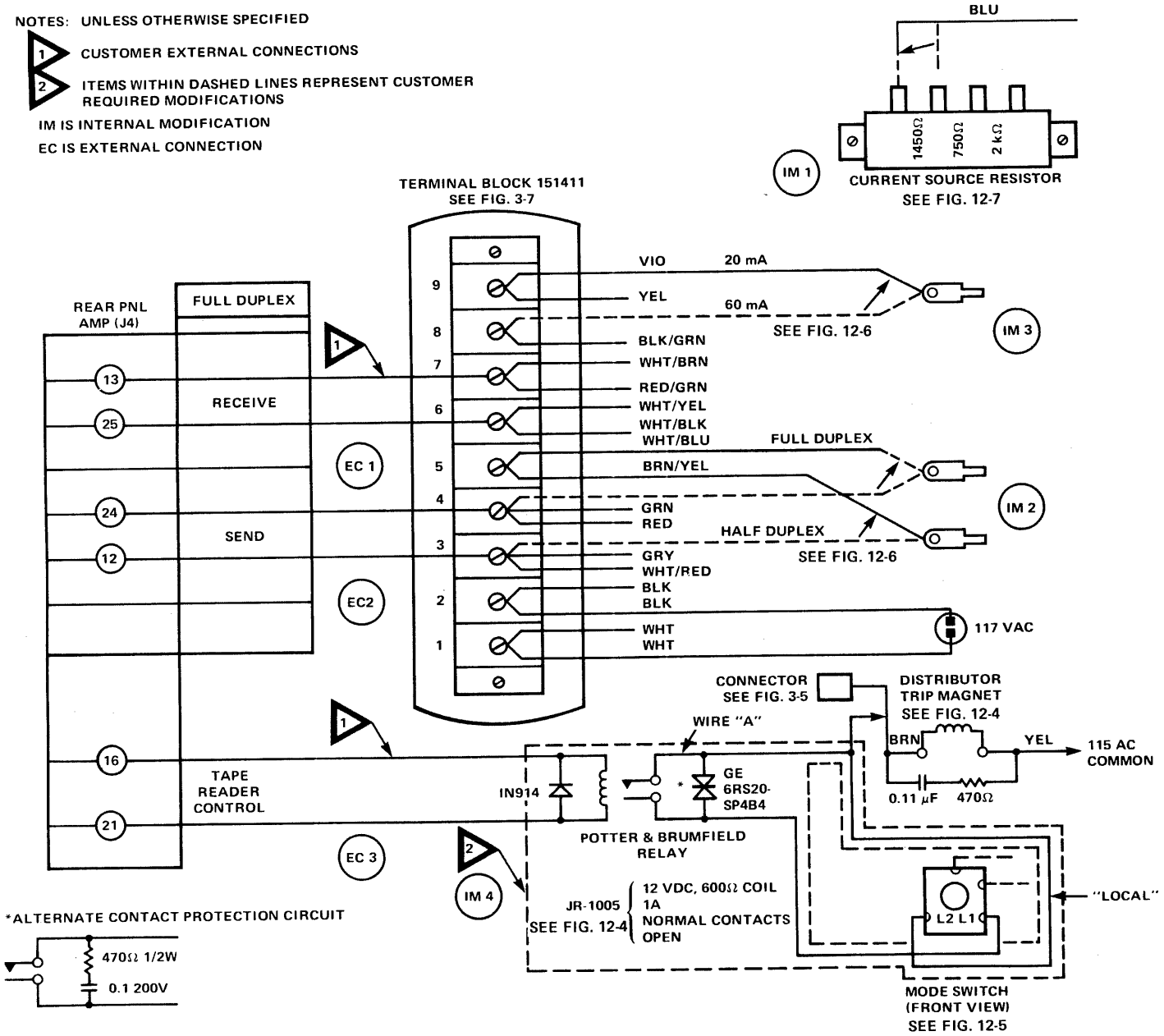


Figure 12-9. Teletype Modification

Appendix A
8080 INSTRUCTION SET SUMMARY

A computer, no matter how sophisticated, can only do what it is "told" to do. One "tells" the computer what to do via a series of coded instructions referred to as a Program. The realm of the programmer is referred to as Software, in contrast to the Hardware that comprises the actual computer equipment. A computer's software refers to all of the programs that have been written for that computer.

When a computer is designed, the engineers provide the Central Processing Unit (CPU) with the ability to perform a particular set of operations. The CPU is designed such that a specific operation is performed when the CPU control logic decodes a particular instruction. Consequently, the operations that can be performed by a CPU define the computer's Instruction Set.

Each computer instruction allows the programmer to initiate the performance of a specific operation. All computers implement certain arithmetic operations in their instruction set, such as an instruction to add the contents of two registers. Often logical operations (e.g., OR the contents of two registers) and register operate instructions (e.g., increment a register) are included in the instruction set. A computer's instruction set will also have instructions that move data between registers, between a register and memory, and between register and an I/O device. Most instruction sets also provide Conditional Instructions. A conditional instruction specifies an operation to be performed only if certain conditions have been met; for example, jump to a particular instruction if the result of the last operation was zero. Conditional instructions provide a program with a decision-making capability.

By logically organizing a sequence of instructions into a coherent program, the programmer can "tell" the computer to perform a very specific and useful function.

The computer, however, can only execute programs whose instructions are in a binary coded form (i.e., a series of 1's and 0's), that is called Machine Code. Because it would be extremely cumbersome to program in machine code, programming languages have been developed. There are programs available which convert the programming language instructions into machine code that can be interpreted by the processor.

One type of programming language is Assembly Language. A unique assembly language mnemonic is assigned to each of the computer's instructions. The programmer can write a program (called the Source Program) using these mnemonics and certain operands; the source program is then converted into machine instructions (called the Object Code). Each assembly language instructions is converted into one machine code instruction (1 or more bytes) by an Assembler program. Assembly languages are usually machine dependent (i.e., they are usually able to run on only one type of computer).

THE 8080 INSTRUCTION SET

The 8080 instruction set includes five different types of instructions:

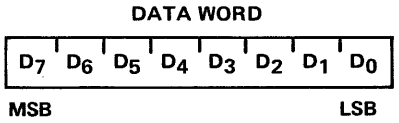
- *Data Transfer Group* – move data between registers or between memory and registers.
- *Arithmetic Group* – add, subtract, increment or decrement data in registers or in memory.
- *Logical Group* – AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory.
- *Branch Group* – conditional and unconditional jump instructions, subroutine call instructions and return instructions.
- *Stack, I/O and Machine Control Group* – includes I/O instructions, as well as instructions for maintaining the stack and internal control flags.

Instruction and Data Formats

Memory for the 8080 is organized into 8-bit quantities, called Bytes. Each byte has a unique 16-bit binary address corresponding to its sequential position in memory.

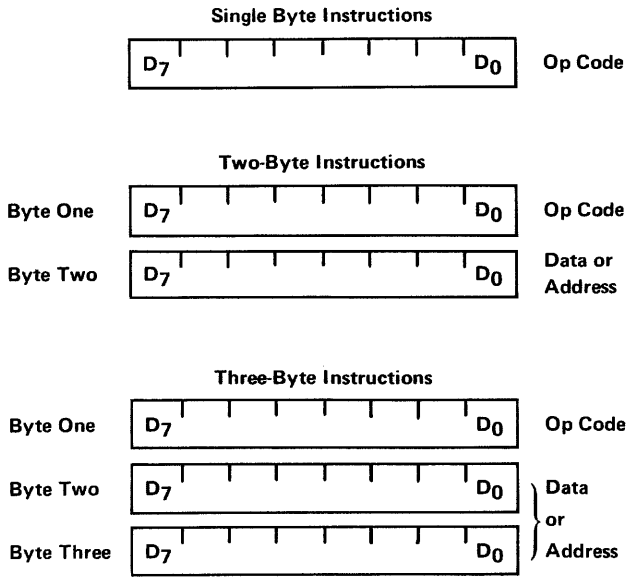
The 8080 can directly address up to 65,536 bytes of memory, which may consist of both read-only memory (ROM) elements and random-access memory (RAM) elements (read/write memory).

Data in the 8080 is stored in the form of 8-bit binary integers:



When a register or data word contains a binary number, it is necessary to establish the order in which the bits of the number are written. In the Intel 8080, BIT 0 is referred to as the Least Significant Bit (LSB), and BIT 7 (of an 8-bit number) is referred to as the Most Significant Bit (MSB).

The 8080 program instructions may be one, two or three bytes in length. Multiple byte instructions must be stored in successive memory locations; the address of the first byte is always used as the address of the instructions. The exact instruction format will depend on the particular operation to be executed.



Addressing Modes

Often the data that is to be operated on is stored in memory. When multi-byte numeric data is used, the data, like instructions, is stored in successive memory locations, with the least significant byte first, followed by increasingly significant bytes. The 8080 has four different modes for addressing data stored in memory or in registers:

- *Direct* – Bytes 2 and 3 of the instruction contain the exact memory address of the data item (the low-order bits of the address are in byte 2, the high-order bits in byte 3).

- *Register* – The instruction specifies the register-pair in which the data is located.

- *Register Indirect* – The instruction specifies a register-pair which contains the memory address where the data is located (the high-order bits of the address are in the first register of the pair, the low-order bits in the second).

- *Immediate* – The instruction contains the data itself. This is either an 8-bit quantity or a 16-bit quantity (least significant byte first, most significant byte second).

Unless directed by an interrupt or branch instruction, the execution of instructions proceeds through consecutively increasing memory locations. A branch instruction can specify the address of the next instruction to be executed in one of two ways:

- *Direct* – The branch instruction contains the address of the next instruction to be executed. (Except for the 'RST' instruction, byte 2 contains the low-order address and byte 3 the high-order address.)

- *Register Indirect* – The branch instruction indicates a register-pair which contains the address of the next instruction to be executed. (The high-order bits of the address are in the first register of the pair, the low-order bits in the second.)

The RST instruction is a special one-byte call instruction (usually used during interrupt sequences).

RST includes a three-bit field; program control is transferred to the instruction whose address is eight times the contents of this three-bit field.

Condition Flags

There are five condition flags associated with the execution of instructions on the 8080. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and are each represented by a 1-bit register in the CPU. A flag is "set" by forcing the bit to 1; "reset" by forcing the bit to 0.

Unless indicated otherwise, when an instruction affects a flag, it affects it in the following manner:

Zero: If the result of an instruction has the value 0, this flag is set; otherwise it is reset.

Sign: If the most significant bit of the result of the operation has the value 1, this flag is set; otherwise it is reset.

Parity: If the modulo 2 sum of the bits of the result of the operation is 0 (i.e., if the result has even parity), this flag is set; otherwise it is reset (i.e., if the result has odd parity).

Carry: If the instruction resulted in a carry (from addition), or a borrow (from subtraction of a comparison) out of the high-order bit, this flag is set; otherwise it is reset.

Auxiliary Carry: If the instruction caused a carry out of bit 3 and into bit 4 of the resulting value, the auxiliary carry is set; otherwise it is reset. This flag is affected by single precision additions, subtractions, increments, decrements, comparisons, and logical operations, but is principally used with additions and increments preceding a DAA (Decimal Adjust Accumulator) instruction.

Symbols and Abbreviations

The following symbols and abbreviations are used in the subsequent description of the 8080 instructions:

SYMBOLS	MEANING
accumulator	Register A
addr	16-bit address quantity
data	8-bit data quantity
data 16	16-bit data quantity
byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r,r1,r2	One of the registers A,B,C,D,E,H,L
DDD,SSS	The bit pattern designating one of the registers A,B,C,D,E,H,L. (DD= destination, SSS=source):

DDD or SSS	REGISTER NAME
111	A
000	B
001	C
010	D
011	E
100	H
101	L

rp	One of the register pairs: B represents the B,C pair with B as the high-order register and C as the low-order register; D represents the D,E pair with D as the high-order register and E as the low-order register; H represents the H,L pair with H as the high-order register and L as the low-order register; SP represents the 16-bit stack pointer register.
RP	The bit pattern designating one of the register pairs B,D,H,SP:

RP	REGISTER PAIR
00	B-C
01	D-E
10	H-L
11	SP

rh	The first (high-order) register of a designated pair.
rl	The second (low-order) register of a designated register pair.

PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits, respectively).
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits, respectively).
r_m	Bit m of the register r (bits are number 7 through 0 from left to right).
Z,S,P,CY,AC	The condition flags: Zero, Sign, Parity, Carry, and Auxiliary Carry, respectively.
( )	The contents of the memory location or registers enclosed in the parentheses.
←	"Is transferred to"A
Λ	Logical AND
⩛	Exclusive OR
V	Inclusive OR
+	Addition
–	Two's complement subtraction
*	Multiplication
↔	"Is exchanged with"
—	The one's complement (e.g., $(\bar{A})$)
n	The restart number 0 through 7
NNN	The binary representation 000 through 111 for restart number 0 through 7, respectively.

Description Format

The following pages provide a detailed description of the instruction set of the 8080. Each instruction is described in the following manner:

1. The MAC 80 assembler format, consisting of the instruction mnemonic and operand fields, is printed in BOLDFACE on the left side of the first line.

2. The name of the instruction is enclosed in parenthesis on the right side of the first line.

3. The next line(s) contain a symbolic description of the operation of the instruction.

4. This is followed by a narative description of the operand of the instruction.

5. The following line(s) contain the binary fields and patterns that comprise the machine instruction.

6. The last four lines contain incidental information about the execution of the instruction. The number of machine cycles and states required to execute the instruction are listed first. If the instruction has two possible execution times, as in a Conditional Jump, both times will be listed, separated by a slash. Next, any significant data addressing modes (see Page A-2) are listed. The last line lists any of the five Flags that are affected by the execution of the instruction.

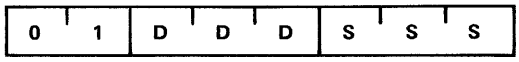
Data Transfer Group

This group of instructions transfer data to and from registers and memory. Condition flags are not affected by any instruction in this group.

MOV r1, r2 (Move Register)

(r1) ← (r2)

The content of register r2 is moved to register r1.

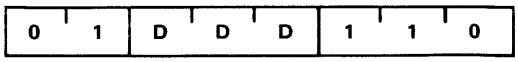


Cycles: 1
States: 5
Addressing: register
Flags: none

MOV r,M (Move from memory)

(r) ← ((H) (L))

The content of the memory location, whose address is in registers H and L, is moved to register r.

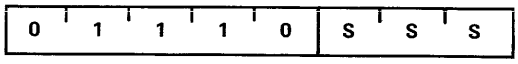


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MOV M, r (Move to memory)

((H)(L)) ← (r)

The content of register r is moved to the memory location whose address is in registers H and L.

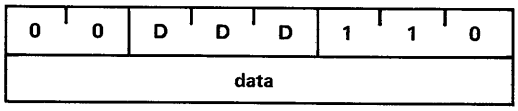


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MVI r, data (Move Immediate)

(r) ← (byte 2)

The content of byte 2 of the instruction is moved to register r.

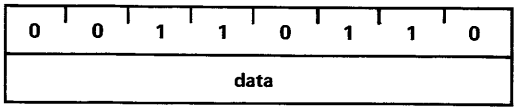


Cycles: 2
States: 7
Addressing: immediate
Flags: none

MVI M, data (Move to memory immediate)

((H)(L)) ← (byte 2)

The content of byte 2 of the instruction is moved to the memory location whose address is in registers H and L.



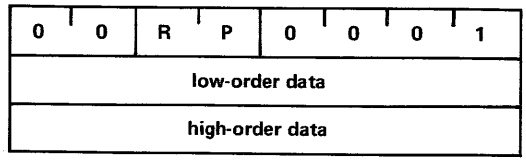
Cycles: 3
States: 10
Addressing: immed./reg. indirect
Flags: none

LXI rp, data 16 (Load register pair immediate)

(rh) ← (byte 3),

(rl) ← (byte 2)

Byte 3 of the instruction is moved into the high-order register (rh) of the register pair rp. Byte 2 of the instruction is moved into the low-order register (rl) of the register pair rp.

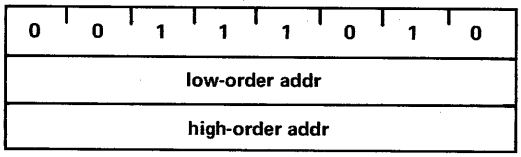


Cycles: 3
States: 10
Addressing: immediate
Flags: none

LDA addr (Load Accumulator direct)

(A) ← ((byte 3)(byte 2))

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register A.

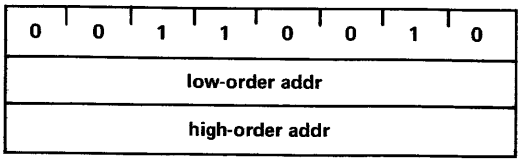


Cycles: 4
States: 13
Addressing: direct
Flags: none

STA addr (Store Accumulator direct)

((byte 3)(byte 2)) ← (A)

The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.



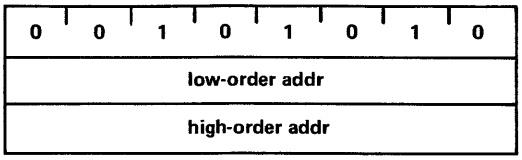
Cycles: 4
States: 13
Addressing: direct
Flags: none

LHLD addr (Load H and L direct)

(L) ← ((byte 3)(byte 2))

(H) ← ((byte 3)(byte 2) + 1)

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register L. The content of the memory location at the succeeding address is moved to register H.



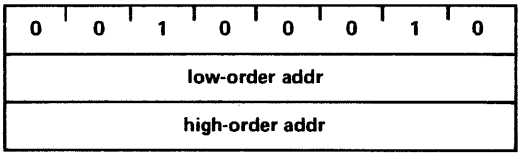
Cycles: 5
States: 16
Addressing: direct
Flags: none

SHLD addr (Store H and L direct)

((byte 3)(byte 2)) ← (L)

((byte 3)(byte 2) + 1) ← (H)

The content of register L is moved to the memory location whose address is specified in byte 2 and byte 3. The content of register H is moved to the succeeding memory location.

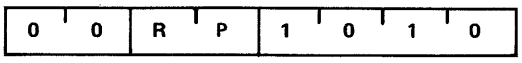


Cycles: 5
States: 16
Addressing: direct
Flags: none

LDAX rp (Load accumulator indirect)

(A) ← ((rp))

The content of the memory location, whose address is in the register pair rp, is moved to register A. Note: Only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

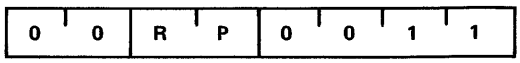


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

STAX rp (Store accumulator indirect)

((rp)) ← (A)

The content of register A is moved to the memory location whose address is in the register pair rp. Note: Only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.



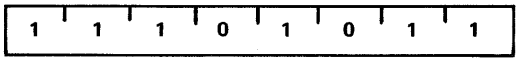
Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

XCHG (Exchange H and L with D and E)

(H) ↔ (D)

(L) ↔ (E)

The contents of registers H and L are exchanged with the contents of registers D and E.

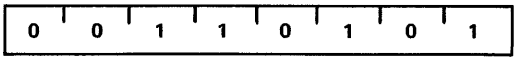


Cycles: 1
States: 4
Addressing: register
Flags: none

DCR M (Decrement memory)

((H)(L)) ← ((H)(L)) – 1

The content of the memory location whose address is contained in the H and L registers is decremented by one. Note: All condition flags except CY are affected.

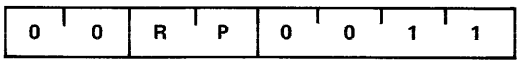


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

INX rp (Increment register pair)

(rh)(rl) ← (rh)(rl) + 1

The content of the register pair rp is incremented by one. Note: No condition flags are affected.



Cycles: 1
States: 5
Addressing: register
Flags: none

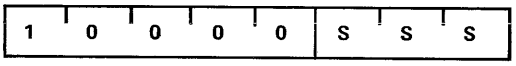
Arithmetic Group

This group of instructions performs arithmetic operations on data in registers and memory.

Unless otherwise indicated, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.

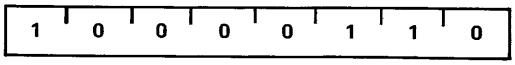
All subtraction operations are performed via two's complement arithmetic, and set the carry flag to one to indicate a borrow and clear it to indicate no borrow.

ADD r (Add Register)
(A) ← (A) + (r)
The content of register r is added to the content of the accumulator. The result is placed in the accumulator.



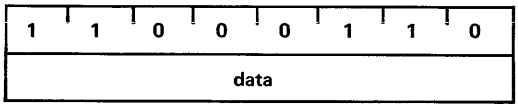
Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADD M (Add Memory)
(A) ← (A) + ((H)(L))
The content of the memory location whose address is contained in the H and L register is added to the content of the accumulator. The result is placed in the accumulator.



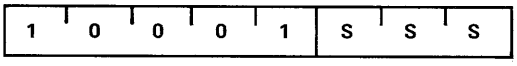
Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ADI data (Add Immediate)
(A) ← (A) + (byte 2)
The content of the second byte of the instruction is added to the constant of the accumulator. The result is placed in the accumulator.



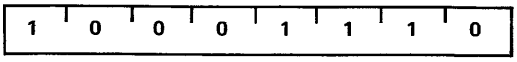
Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ADC r (Add Register with Carry)
(A) ← (A) + (r) + (CY)
The content of register r and the content of the carry bit are added to the content of the accumulator. The result is placed in the accumulator.



Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADC M (Add Memory with Carry)
(A) ← (A) + ((H)(L)) + (CY)
The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are added to the accumulator. The result is placed in the accumulator.

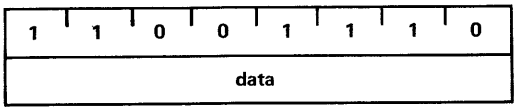


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ACI data (Add Immediate with Carry)

(A) ← (A) + (byte 2) + (CY)

The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator.

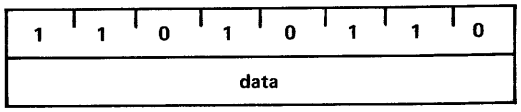


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUI data (Subtract Immediate)

(A) ← (A) – (byte 2)

The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator.

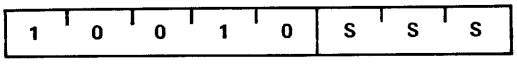


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUB r (Subtract Register)

(A) ← (A) – (r)

The content of register r is subtracted from the content of the accumulator. The result is placed in the accumulator.

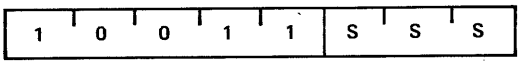


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SBB r (Subtract Register with Borrow)

(A) ← (A) – (r) – (CY)

The content of register r and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

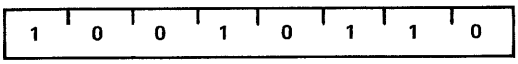


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SUB M (Subtract Memory)

(A) ← (A) – ((H)(L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the content of the accumulator. The result is placed in the accumulator.

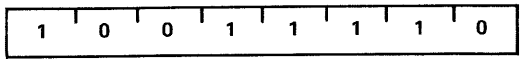


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SBB M (Subtract Memory with Borrow)

(A) ← (A) – ((H)(L)) – (CY)

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

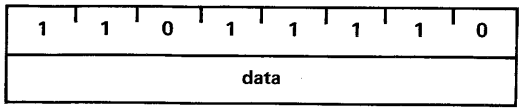


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SBI data (Subtract Immediate with Borrow)

(A) ← (A) – (byte 2) – (CY)

The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

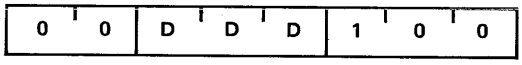


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

INR r (Increment Register)

(r) ← (r) + 1

The content of register r is incremented by one. Note: All condition flags except CY are affected.

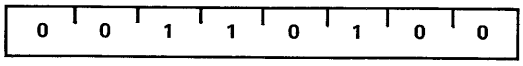


Cycles: 1
States: 5
Addressing: register
Flags: Z,S,P,AC

INR M (Increment Memory)

((H)(L)) ← ((H)(L)) + 1

The content of the memory location whose address is contained in the H and L registers is incremented by one. Note: All condition flags except CY are affected.

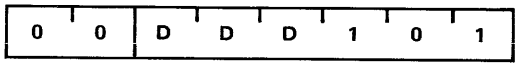


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

DCR r (Decrement Register)

(r) ← (r) – 1

The content of register r is decremented by one. Note: All condition flags except CY are affected.

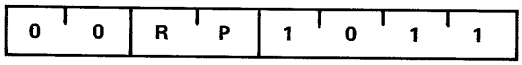


Cycles: 1
States: 5
Addressing: register
Flags: Z,S,P,AC

DCX rp (Decrement register pair)

(rh)(rl) ← (rh)(rl) – 1

The content of the register pair rp is decremented by one. Note: No condition flags are affected.

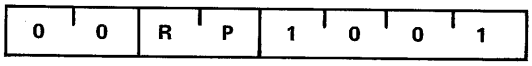


Cycles: 1
States: 5
Addressing: register
Flags: none

DAD rp (Add register pair to H and L)

(H)(L) ← (H)(L) + (rh)(rl)

The content of the register pair rp is added to the content of the register pair H and L. The result is placed in the register pair H and L. Note: Only the CY flag is affected. It is set if there is a carry out of the double precision add; otherwise it is reset.



Cycles: 3
States: 10
Addressing: register
Flags: CY

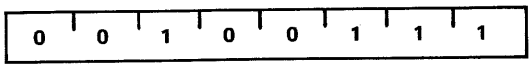
DAA (Decimal Adjust Accumulator)

The 8-bit number in the accumulator is adjusted to form two 4-bit Binary-Coded-Decimal digits by the following process:

1. If the value of the least significant 4 bits of the accumulator is greater than 9 or if the AC flag is set, 6 is added to the accumulator.

2. If the value of the most significant 4 bits of the accumulator is now greater than 9, or if the CY flag is set, 6 is added to the most significant 4 bits of the accumulator.

NOTE: All flags are affected.



Cycles: 1
States: 4
Flags: Z,S,P,CY,AC

Logical Group

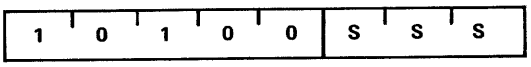
This group of instructions performs logical (Boolean) operations on data in registers and memory and on condition flags.

Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Auxiliary Carry, and Carry flags according to the standard rules.

ANA r (AND Register)

(A) ← (A) ∧ (r)

The content of register r is logically ANDed with the content of the accumulator. The result is placed in the accumulator. The CY flag is cleared.

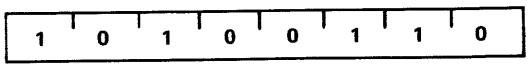


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ANA M (AND memory)

(A) ← (A) ∧ ((H)(L))

The contents of the memory location whose address is contained in the H and L registers is logically ANDed with the content of the accumulator. The CY flag is cleared.

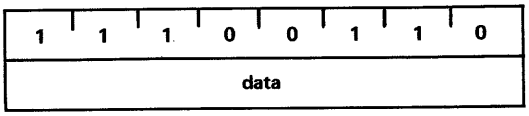


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ANI data (AND immediate)

(A) ← (A) ∧ (byte 2)

The content of the second byte of the instruction is logically ANDed with the contents of the accumulator. The result is placed in the accumulator. The CY flag is cleared.

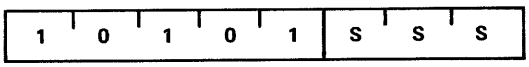


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

XRA r (Exclusive OR Register)

(A) ← (A) ∀ (r)

The content of register r is exclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

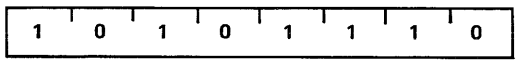


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

XRA M (Exclusive OR Memory)

(A) ← (A) ∀ ((H)(L))

The content of the memory location whose address is contained in the H and L registers is exclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

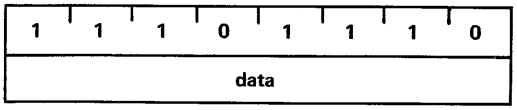


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XRI data (Exclusive OR immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is exclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

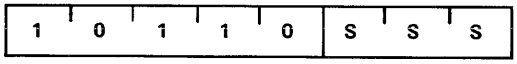


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ORA r (OR Register)

(A) ← (A) V (r)

The content of register r is inclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

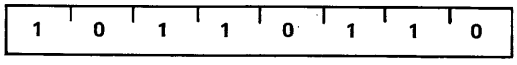


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ORA M (OR Memory)

(A) ← (A) V ((H)(L))

The content of the memory location whose address is contained in the H and L registers is inclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

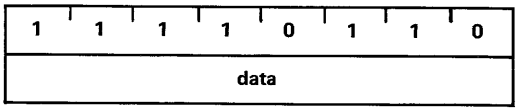


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ORI data (OR Immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is inclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

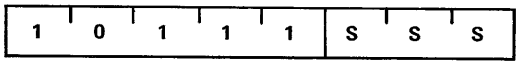


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

CMP r (Compare Register)

(A) – (r)

The content of register r is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if (A) = (r). The CY flag is set to 1 if (A) < (r).

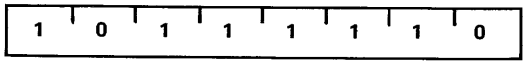


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

CMP M (Compare memory)

(A) – ((H)(L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 of (A) = ((H)(L)). The CY flag is set to 1 if (A) < ((H)(L)).

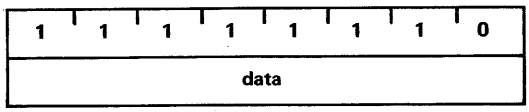


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

CPI data (Compare immediate)

(A) – (byte 2)

The content of the second byte of the instruction is subtracted from the accumulator. The condition flags are set by the result of the subtraction. The Z flag is set to 1 if (A) = (byte 2). The CY flag is set to 1 if (A) < (byte 2).



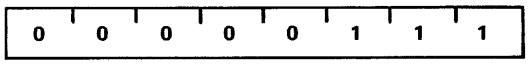
Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

RLC (Rotate left)

$(A_{n+1}) \leftarrow (A_n)$; $(A_0) \leftarrow (A_7)$

$(CY) \leftarrow (A_7)$

The content of the accumulator is rotated left one position. The low-order bits and the CY flag are both set to the value shifted out of the high-order bit position. Only the CY flag is affected.



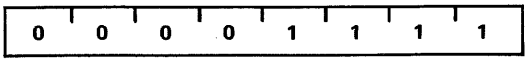
Cycles: 1
States: 4
Flags: CY

RRC (Rotate right)

$(A_n) \leftarrow (A_{n-1})$; $(A_7) \leftarrow (A_0)$

$(CY) \leftarrow (A_0)$

The content of the accumulator is rotated right one position. The high-order bit and the CY flag are both set to the value shifted out of the low-order bit position. Only the CY flag is affected.



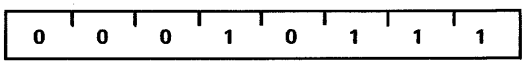
Cycles: 1
States: 4
Flags: CY

RAL (Rotate left through carry)

$(A_{n+1}) \leftarrow (A_n)$; $(CY) \leftarrow (A_7)$

$(A_0) \leftarrow (CY)$

The content of the accumulator is rotated left one position through the CY flag. The low-order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high-order bit. Only the CY flag is affected.



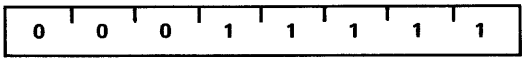
Cycles: 1
States: 4
Flags: CY

RAR (Rotate right through carry)

$(A_n) \leftarrow (A_{n+1})$; $(CY) \leftarrow (A_0)$

$(A_7) \leftarrow (CY)$

The content of the accumulator is rotated right one position through the CY flag. The high-order bit is set to the CY flag and the CY flag is set to the value shifted out of the low-order bit. Only the CY flag is affected.

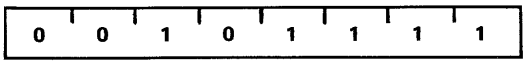


Cycles: 1
States: 4
Flags: CY

CMA (Complement accumulator)

$(A) \leftarrow (A)$

The contents of the accumulator are complemented (zero bits become 1, one bits become 0). No flags are affected.

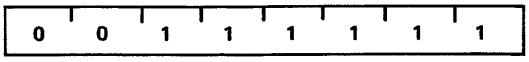


Cycles: 1
States: 4
Flags: none

CMC (Complement carry)

$(CY) \leftarrow (CY)$

The CY flag is complemented. No other flags are affected.

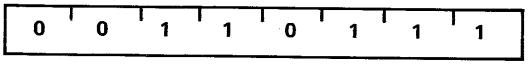


Cycles: 1
States: 4
Flags: CY

STC (Set carry)

(CY) ← 1

The CY flag is set to 1. No other flags are affected.



Cycles: 1
States: 4
Flags: CY

Branch Group

This group of instructions alter normal sequential program flow.

Condition flags are not affected by an instruction in this group.

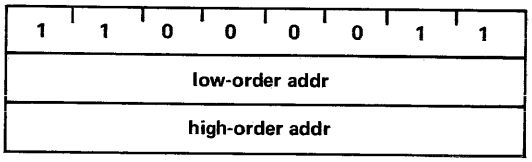
The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed. The conditions that may be specified are as follows:

CONDITION		CCC
NZ	– not zero (Z=0)	000
Z	– zero (Z = 1)	001
NC	– no carry (C = 0)	010
C	– carry (CY = 1)	011
PO	– parity odd (P = 0)	100
PE	– parity even (P = 1)	101
P	– plus (S = 0)	110
M	– minus (S = 1)	111

JMP addr (Jump)

(PC) → (byte 3)(byte 2)

Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.



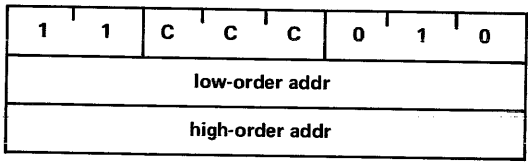
Cycles: 3
States: 10
Addressing: immediate
Flags: none

Jcondition addr (Conditional jump)

If (CCC),

(PC) ← (byte 3)(byte 2)

If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially.



Cycles: 3
States: 10
Addressing: immediate
Flags: none

CALL addr (Call)

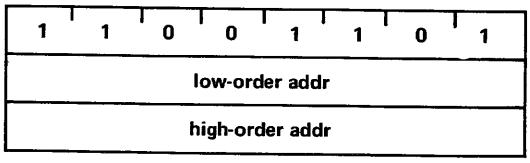
((SP) – 1) ← (PCH)

((SP) – 2) ← (PCL)

(SP) ← (SP) – 2

(PC) ← (byte 3)(byte 2)

The high-order 8 bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order 8 bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

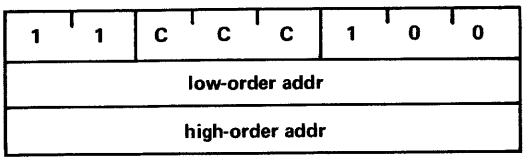


Cycles: 5
States: 17
Addressing: immed./reg. indirect
Flags: none

Ccondition addr (Condition call)

If (CCC),
((SP) – 1) ← (PCH)
((SP) – 2) ← (PCL)
(SP) ← (SP) – 2
(PC) ← (byte 3)(byte 2)

If the specified condition is true, the actions specified in the CALL instruction (see above) are performed; otherwise, control continues sequentially.

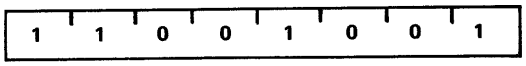


Cycles: 3/5
States: 11/17
Addressing: immed./reg. indirect
Flags: none

RET (Return)

(PCL) ← ((SP));
(PCH) ← ((SP) + 1);
(SP) ← (SP) + 2;

The content of the memory location whose address is specified in register SP is moved to the low-order 8 bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order 8 bits of register PC. The content of register SP is incremented by 2.

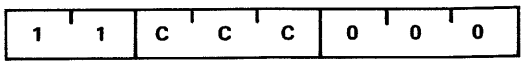


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none

Rcondition (Conditional return)

If (CCC),
(PCL) ← ((SP))
(PCH) ← ((SP) + 1)
(SP) ← (SP) + 2

If the specified condition is true, the actions specified in the RET instruction (see above) are performed; otherwise, control continues sequentially.

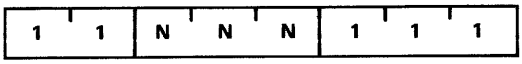


Cycles: 1/3
States: 5/11
Addressing: reg. indirect
Flags: none

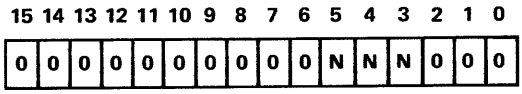
RST n (Restart)

((SP) – 1) ← (PCH)
((SP) – 2) ← (PCL)
(SP) ← (SP) – 2
(PC) ← 8 * (NNN)

The high-order 8 bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order 8 bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is eight times the content of NNN.



Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

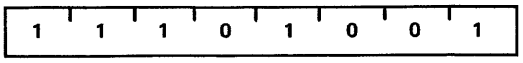


Program Counter After Restart

PCHL (Jump H and L indirect – move H and L to PC)

(PCH) ← (H)
(PCL) ← (L)

The content of register H is moved to the high-order 8 bits of register PC. The content of register L is moved to the low-order 8 bits of register PC.



Cycles: 1
States: 5
Addressing: register
Flags: none

Stack, I/O, and Machine Control Group

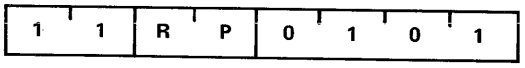
This group of instructions performs I/O, manipulates the Stack, and alters internal control flags.

Unless otherwise specified, condition flags are not affected by any instructions in this group.

PUSH rp (Push)

((SP) – 1) ← (rh)
((SP) – 2) ← (rl)
(SP) ← (SP) – 2

The content of the high-order register of register pair rp is moved to the memory location whose address is one less than the content of register SP. The content of the low-order register of register pair rp is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Note: Register pair rp=SP may not be specified.

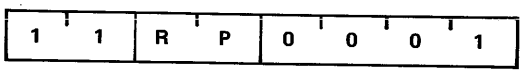


Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

POP rp (Pop)

(rl) ← ((SP))
(rh) ← ((SP) + 1)
(SP) ← (SP) + 2

The content of the memory location, whose address is specified by the content of register SP, is moved to the low-order register of register pair rp. The content of the memory location, whose address is one more than the content of register SP, is moved to the high-order register of register pair rp. The content of register SP is incremented by 2. Note: Register pair rp=SP may not be specified.

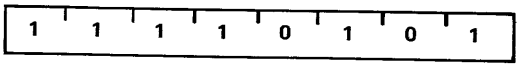


Cycles: 3
States: 10
Addressing: reg. indirect.
Flags: none

PUSH PSW (Push processor status word)

((SP) – 1) ← (A)
((SP) – 2)$_0$ ← (CY, ((SP) – 2)$_1$ ← 1
((SP) – 2)$_2$ ← (P), ((SP) – 2)$_3$ ← 0
((SP) – 2)$_4$ ← (AC), ((SP) – 2)$_5$ ← 0
((SP) – 2)$_6$ ← (Z), ((SP) – 2)$_7$ ← (S)
(SP) ← (SP) – 2

The content of register A is moved to the memory location whose address is one less than register SP. The contents of the condition flags are assembled into a processor status word and the word is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two.

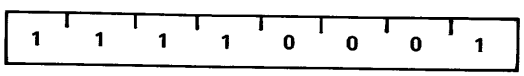


Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

POP PSW (Pop processor status word)

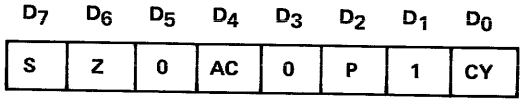
(CY) ← ((SP))$_0$
(P) ← ((SP))$_2$
(AC) ← ((SP))$_4$
(Z) ← ((SP))$_6$
(S) ← ((SP))$_7$
(A) ← ((SP) + 1)
(SP) ← (SP) + 2

The content of the memory location whose address is specified by the content of register SP is used to restore the condition flags. The content of the memory location whose address is one more than the content of register SP is moved to register A. The content of register SP is incremented by 2.



Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

FLAG WORD

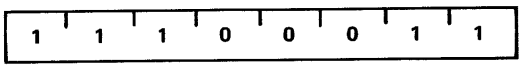


XTHL (Exchange stack top with H and L)

(L) ↔ ((SP))

(H) ↔ ((SP) + 1)

The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

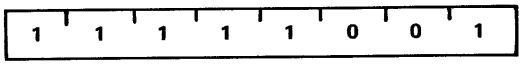


Cycles: 5
States: 18
Addressing: reg. indirect
Flags: none

SPHL (Move HL to SP)

(SP) ← (H)(L)

The contents of registers H and L (16 bits) are moved to register SP.

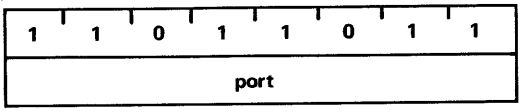


Cycles: 1
States: 5
Addressing: register
Flags: none

IN port (Input)

(A) ← (data)

The data placed on the 8-bit bidirectional data bus by the specified port is moved to register A.

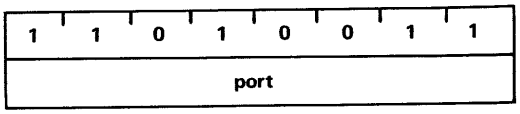


Cycles: 3
States: 10
Addressing: direct
Flags: none

OUT port (Output)

(data) ← (A)

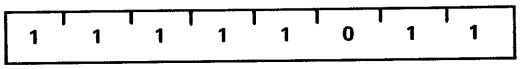
The content of register A is placed on the 8-bit bidirectional data bus for transmission to the specified port.



Cycles: 3
States: 10
Addressing: direct
Flags: none

EI (Enable interrupt)

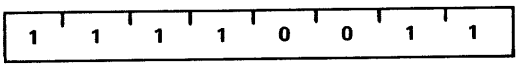
The interrupt system is enabled following the execution of the next instruction.



Cycles: 1
States: 4
Flags: none

DI (Disable interrupts)

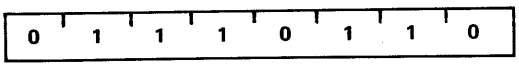
The interrupt system is disabled immediately following the execution of the DI instruction.



Cycles: 1
States: 4
Flags: none

HLT (Halt)

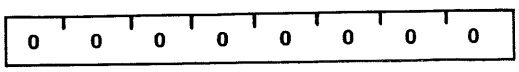
The processor is stopped. The registers and flags are unaffected.



Cycles: 1
States: 7
Flags: none

NOP (No op)

No operation is performed. The registers and flags are unaffected.



Cycles: 1
States: 4
Flags: none

INSTRUCTION SET

Summary of Processor Instructions

MNEMONIC	DESCRIPTION	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	CLOCK[2] CYCLES
$MOV_{r1,r2}$	Move register to register	0	1	D	D	D	S	S	S	5
MOV M,r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r,M	Move memory to register	0	1	D	D	D	1	1	0	7
HLT	Halt	0	1	1	1	0	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
INR r	Increment register	0	0	D	D	D	1	0	0	5
DCR r	Decrement register	0	0	D	D	D	1	0	1	5
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10
JZ	Jump on zero	1	1	0	0	1	0	1	0	10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10
JP	Jump on positive	1	1	1	1	0	0	1	0	10
JM	Jump on minus	1	1	1	1	1	0	1	0	10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10
CALL	Call unconditional	1	1	0	0	1	1	0	1	17
CC	Call on carry	1	1	0	1	1	1	0	0	11/17
CNC	Call on no carry	1	1	0	1	0	1	0	0	11/17
CZ	Call on zero	1	1	0	0	1	1	0	0	11/17
CNZ	Call on no zero	1	1	0	0	0	1	0	0	11/17
CP	Call on positive	1	1	1	1	0	1	0	0	11/17
CM	Call on minus	1	1	1	1	1	1	0	0	11/17
CPE	Call on parity even	1	1	1	0	1	1	0	0	11/17
CPO	Call on parity odd	1	1	1	0	0	1	0	0	11/17
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	5/11
RNC	Return on no carry	1	1	0	1	0	0	0	0	5/11

MNEMONIC	DESCRIPTION	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	CLOCK[2] CYCLES
RZ	Return on zero	1	1	0	0	1	0	0	0	5/11
RNZ	Return on no zero	1	1	0	0	0	0	0	0	5/11
RP	Return on positive	1	1	1	1	0	0	0	0	5/11
RM	Return on minus	1	1	1	1	1	0	0	0	5/11
RPE	Return on parity even	1	1	1	0	1	0	0	0	5/11
RPO	Return on parity odd	1	1	1	0	0	0	0	0	5/11
RST	Restart	1	1	A	A	A	1	1	1	11
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	11
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	11
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	11
POP B	Pop register pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
XTHL	Exchange top of stock H & L	1	1	1	0	0	0	1	1	18
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	5
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	5
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	5
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	5
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	5
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	5
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	5
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4

NOTES: 1. DDD or SSS – 000 B – 001 C – 010 D – 011 E – 100 H – 101 L – 110 Memory – 111 A.
2. Two possible cycle times, (5/11) indicate instruction cycles dependent on condition flags.

Appendix B
ASCII TABLE

The INTELLEC MDS uses a 7-bit ASCII code, which is the normal 8-bit ASCII code with the parity (high order) bit always reset.

GRAPHIC OR CONTROL	ASCII (HEXADECIMAL)	GRAPHIC OR CONTROL	ASCII (HEXADECIMAL)
NULL	00	ACK	7C
SOM	01	Alt. Mode	7D
EOA	02	Rubout	7F
EOM	03	!	21
EOT	04	"	22
WRU	05	#	23
RU	06	$	24
BELL	07	%	25
FE	08	&	26
H. Tab	09	'	27
Line Feed	0A	(	28
V. Tab	0B	)	29
Form	0C	*	2A
Return	0D	+	2B
SO	0E	,	2C
SI	0F	–	2D
DCO	10	.	2E
X-On	11	/	2F
Tape Aux. On	12	:	3A
X-Off	13	;	3B
Tape Aux. Off	14	<	3C
Error	15	=	3D
Sync	16	>	3E
LEM	17	?	3F
SO	18	[	5B
S1	19	/	5C
S2	1A	]	5D
S3	1B	↑	5E
S4	1C	←	5F
S5	1D	@	40
S6	1E	blank	20
S7	1F	0	30

Appendix C
BINARY–DECIMAL–HEXADECIMAL CONVERSION TABLES

HEXADECIMAL ARITHMETIC

ADDITION TABLE

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
2	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
3	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12
4	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
5	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14
6	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15
7	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16
8	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17
9	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18
A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A
C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

MULTIPLICATION TABLE

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
3	06	09	0C	0F	12	15	18	1B	1E	21	24	27	2A	2D
4	08	0C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0A	0F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	14	1E	28	32	30	46	50	5A	64	6E	78	82	8C	96
B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	1E	2D	3C	48	5A	69	78	87	96	A5	B4	C3	D2	E1

POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 347 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 916 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 726 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 521 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 688 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
562 940 953 421 213	49	0.000 000 000 000 001 776 866 839 499 259 464 677 810 668 945 312 5
1 125 899 906 842 624	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
2 251 799 813 685 248	51	0.000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
4 503 599 627 370 496	52	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25
18 014 398 509 481 984	54	0.000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
36 028 797 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5
72 057 594 037 927 936	56	0.000 000 000 000 000 013 877 787 807 814 456 755 295 395 851 135 253 906 25
144 115 188 075 855 872	57	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 925 567 676 950 125
288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
1 152 921 504 606 846 976	60	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
2 305 843 009 213 693 952	61	0.000 000 000 000 000 000 433 680 868 994 201 773 602 981 120 347 976 684 570 312 5
4 611 686 018 427 387 904	62	0.000 000 000 000 000 000 216 840 434 497 100 886 801 490 560 173 988 342 285 156 25
9 223 372 036 854 775 808	63	0.000 000 000 000 000 000 108 420 217 248 550 443 400 745 280 086 994 171 142 578 125

TABLE OF POWERS OF SIXTEEN$_{10}$

16^n_n	n	16^{-n}
++1	0	0.10000 00000 00000 00000 X 10
16	1	0.62500 00000 00000 00000 X 10^{-1}
256	2	0.39062 50000 00000 00000 X 10^{-2}
4 096	3	0.24414 06250 00000 00000 X 10^{-3}
65 536	4	0.15258 78906 25000 00000 X 10^{-4}
1 048 576	5	0.95367 43164 06250 00000 X 10^{-6}
16 777 216	6	0.59604 64477 53906 25000 X 10^{-7}
268 435 456	7	0.37252 90298 46191 40625 X 10^{-8}
4 294 967 296	8	0.23283 06436 53869 62891 X 10^{-9}
68 719 476 736	9	0.14551 91522 83668 51807 X 10^{-10}
1 099 511 627 776	10	0.90949 47017 72928 23792 X 10^{-12}
17 592 186 044 416	11	0.56843 41886 08080 14870 X 10^{-13}
281 474 976 710 656	12	0.35527 13678 80050 09294 X 10^{-14}
4 503 599 627 370 496	13	0.22204 46049 25031 30808 X 10^{-15}
72 057 594 037 927 936	14	0.13877 78780 78144 56755 X 10^{-16}
1 152 921 504 606 846 976	15	0.86736 17379 88403 54721 X 10^{-18}

TABLE OF POWERS OF 10_{16}

10^n	n	10^{-n}
1	0	1.0000 0000 0000 0000
A	1	0.1999 9999 9999 999A
64	2	0.28F5 C28F 5C28 F5C3 X 16^{-1}
3E8	3	0.4189 374B C6A7 EF9E X 16^{-2}
2710	4	0.68DB 8BAC 710C B290 X 16^{-3}
86A0	5	0 A7C5 AC47 1B47 8423 X 16^{-4}
4240	6	0.10C7 F7A0 B5ED 8D37 X 16^{-4}
9680	7	0.1AD7 F29A BCAF 4858 X 16^{-5}
E100	8	0.2AF3 1DC4 6118 73BF X 16^{-6}
CA00	9	0.44B8 2FA0 9B5A 52CC X 16^{-7}
540B E400	10	0.6DF3 7F67 SEF0 EADF X 16^{-8}
4876 E800	11	0.AFEB FF0B CB24 AAFF X 16^{-9}
D4A5 1000	12	0.1197 9981 2DEA 1119 X 16^{-9}
4E72 A000	13	0.1025 C268 4976 81C2 X 16^{-10}
107A 4000	14	0.2D09 370D 4257 3604 X 16^{-11}
8D7E A4C6 3000	15	0.480E BE7B 9D58 566D X 16^{-12}
8652 6FC1 0000	16	0.734A CA5F 6226 F0AE X 16^{-13}
4578 5D8A 0000	17	0.B877 AA32 36A4 B449 X 16^{-14}
B6B3 A764 0000	18	0.1272 5DD1 D243 ABA1 X 16^{-15}
2304 89E8 0000	19	0.1D83 C94F B6D2 AC35 X 16^{-15}

HEXADECIMAL-DECIMAL INTEGER CONVERSION

The table below provides for direct conversions between hexadecimal integers in the range 0–FFF and decimal integers in the range 0–4095. For conversions of larger integers, the table values may be added to the following figures:

HEXADECIMAL	DECIMAL	HEXADECIMAL	DECIMAL
01 000	4 096	20 000	131 072
02 000	8 192	30 000	196 608
03 000	12 288	40 000	262 144
04 000	16 384	50 000	327 680
05 000	20 480	60 000	393 216
06 000	24 576	70 000	458 752
07 000	28 672	80 000	524 288
08 000	32 768	90 000	589 824
09 000	36 864	A0 000	655 360
0A 000	40 960	B0 000	720 896
0B 000	45 056	C0 000	786 432
0C 000	49 152	D0 000	851 968
0D 000	53 248	E0 000	917 504
0E 000	57 344	F0 000	933 040
0F 000	61 440	100 000	1 048 576
10 000	65 536	200 000	2 097 152
11 000	69 632	300 000	3 145 728
12 000	73 728	400 000	4 194 304
13 000	77 824	500 000	5 242 880
14 000	81 920	600 000	6 291 456
15 000	86 016	700 000	7 340 032
16 000	90 112	800 000	8 388 608
17 000	94 208	900 000	9 437 184
18 000	98 304	A00 000	10 485 760
19 000	102 400	B00 000	11 534 336
1A 000	106 496	C00 000	12 582 912
1B 000	110 592	D00 000	13 631 489
1C 000	114 638	E00 000	14 680 064
1D 000	118 784	F00 000	15 728 640
1E 000	122 880	1 000 000	16 777 216
1F 000	126 976	2 000 000	33 554 432

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
010	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
020	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
030	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
040	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
050	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
060	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
070	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
080	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
090	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	1055	0156	0157	0158	0159
0A0	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B0	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C0	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	2007
0D0	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E0	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F0	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255

HEXADECIMAL-DECIMAL INTEGER CONVERSION (continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
100	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
110	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
120	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
130	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
140	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0331	0333	0334	0335
150	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
160	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
170	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
180	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
190	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A0	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B0	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C0	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D0	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E0	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F0	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511
200	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
210	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
220	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
230	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
240	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
250	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
260	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
270	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
280	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
290	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A0	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B0	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C0	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D0	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E0	0736	0738	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F0	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
300	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
310	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
320	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
330	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
340	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
350	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
360	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
370	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
380	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
390	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A0	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B0	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C0	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D0	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E0	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F0	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

HEXADECIMAL-DECIMAL INTEGER CONVERSION (continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
400	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
410	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
420	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
430	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
440	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
450	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
460	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
470	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
480	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1163	1165	1166	1167
490	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A0	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B0	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C0	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D0	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E0	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1263	1263
4F0	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
500	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
510	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
520	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
530	1382	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
540	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
550	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	11372	1373	1374	1375
560	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
570	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
580	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
590	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A0	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B0	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C0	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D0	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E0	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F0	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535
600	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
610	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
620	1568	1569	1570	1571	1572	1573	1574	1575	1576n	1577	1578	1579	1580	1581	1582	1583
630	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
640	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
650	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
660	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
670	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
680	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
690	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A0	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B0	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C0	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D0	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E0	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F0	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791

HEXADECIMAL-DECIMAL INTEGER CONVERSION (continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
700	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
710	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
720	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
730	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
740	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
750	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
760	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
770	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
780	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
790	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A0	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B0	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C0	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D0	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E0	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F0	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047
800	2048	2049	2050	2051	2502	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
810	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
820	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
830	2096	2097	2098	2099	2100	2101	2102	2103	2104	2015	2106	2107	2108	2109	2110	2111
840	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
850	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
860	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
870	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
880	2176	21777	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
890	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A0	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B0	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C0	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D0	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E0	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F0	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
900	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
910	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
920	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
930	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
940	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
950	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
960	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
970	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
980	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
990	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A0	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B0	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C0	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D0	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E0	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F0	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

HEXADECIMAL-DECIMAL INTEGER CONVERSION (continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A00	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A10	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A20	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A30	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A40	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A50	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A60	2656	2657	2658	2659	2660	2661	2622	2663	2664	2665	2666	2667	2668	2669	2670	2671
A70	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A80	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A90	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA0	2720	2721	2722	2723	2724	2725	2726	27272	2728	2729	2730	2731	2732	2733	2734	2735
AB0	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC0	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD0	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE0	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF0	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B00	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B10	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B20	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B30	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B40	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B50	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B60	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B70	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B80	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B90	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA0	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB0	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC0	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD0	3024	3025	3026	3027	3028	3029	3030	3031	3032	3933	3034	3035	3036	3037	3038	3039
BE0	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF0	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071
C00	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C10	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C20	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C30	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C40	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C50	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C60	3168	3168	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C70	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C80	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C90	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA0	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB0	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC0	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD0	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE0	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF0	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327

HEXADECIMAL-DECIMAL INTEGER CONVERSION (continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D00	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D10	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D20	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D30	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D40	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D50	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D60	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D70	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D80	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D90	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA0	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB0	3504	3505	3506	3507	3508	3509	3410	3511	3512	3513	3514	3515	1516	3517	3518	3519
DC0	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
CC0	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE0	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF0	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583
E00	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E10	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E20	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E30	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E40	3648	3648	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E50	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E60	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E70	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3070	3708	3709	3710	3711
E80	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E90	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA0	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB0	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC0	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED0	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE0	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF0	3824	3825	3826	3827	3828	3829	3030	3831	3832	3833	3834	3835	3836	3837	3838	3839
F00	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F10	3856	3857	3858	3859	3860	3861	3862	3863	3865	3865	3866	3867	3868	3869	3870	3871
F20	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F30	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F40	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F50	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F60	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F70	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F80	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F90	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA0	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB0	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC0	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD0	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE0	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF0	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 • (408) 987-8080

Printed in U.S.A. A3/179/2K NCG